ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES
COLLEGE OF MANAGEMENT, INFORMATION AND ECONOMIC SCIENCE
SCHOOL OF INFORMATION SCIENCE

Concatenative Text-To-Speech System for Afaan Oromo Language

BY

Samson Taddese Ofgaa

A Thesis Submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirements for the Degree of Masters of Science in Information Science

May, 2011
AAU

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

COLLEGE OF MANAGEMENT, INFORMATION AND ECONOMIC SCIENCE

SCHOOL OF INFORMATION SCIENCE

CONCATENATIVE TEXT-TO-SPEECH SYSTEM FOR AFAAN OROMOO LANGUAGE

BY

Samson Taddese Ofgaa

Signature of Board of Examiners for Approval

Ato Mulu G/Egziabher (MSc.) _____ _____

_____ _____

_____ _____

_____ _____

Dedication

To Mom & Dad

Acknowledgment

First of all I would like to thank my God; my savior and redeemer. Then my special thanks goes to my adviser Ato Mullu G/egziabher who has been a right hand with a constructive ideas and comments from the very beginning of this work to the end.

I am also indebted to Ato Biniam Efrem and Ato Tsegaye W/Mariam from linguistics department of Addis Ababa University for their help during the Afaan Oromo corpus collections.

I also would like to appreciate the enthusiasm I got from the Informatics faculty specifically the information science department in financing the thesis work conducted.

Oh, where would this work and I be if intimate friends like Digajara, Nati and Kal were not around during those days with full of stress?

Finally, my deepest gratitude's are for my family for their inevitable prayers, love and encouragements they showed during my lengthy stay alone at night. I love you all!

Abstract

This paper explores the possibility of developing a concatenative TTS system for Afaan Oromo language where diphone and triphones are the speech units that are focused on.

Nowadays, concatenative method is used in most modern TTS systems to produce synthesized speech. But in concatenative method, selecting an appropriate unit for creating a database is a challenging task. In the proposed approach, such database is created with different sizes of speech units and is used to produce speech utterances which include diphones and triphones. For the synthesis process, diphones and triphones which are smaller speech units are used to achieve unlimited vocabulary of speech.

During the process, a diphone database consisting of 800 entries and a triphone database with entries 1982 is constructed. The synthesizer is then evaluated for its performance measure, naturalness and intelligence by six individuals from the language domain.

The experimental results show that 75% and 54% of words in the data set are correctly pronounced as to the diphone and triphone speech units, respectively. The MOS levels of the intelligence of the system also showed that a 3.03 and 2.2 scale levels were achieved for the diphone and triphones respectively; whereas the naturalness of the system was 2.65 and 2.02 for each speech units respectively. The removal of many triphone speech units

that can increase the time complexity of the system and those that don't represent the language can be mentioned as the main reason behind the low result of the triphones as compared to the diphones.

In fact, the values gained for the triphones has shown an increase from 2.2 to 2.23 and then to 2.27 for the measured systems intelligence and from 2.02 to 2.05 then to 2.08 for naturalness of the system when some of the removed entries are added to the database. The result obtained indeed is a promising result; for which accordingly, future research directions are proposed to improve the performance of the system.

Key words: Speech Synthesis, Concatenative methods, Festival, Afaan Oromo

Table Of Contents

List of Tables

List of Figures

List of Acronyms

CSTR	Center for Speech Technology Research
DM	Delta Modulation
DPCM	Differential Pulse Code Modulation
DRT	Diagnostic Rhyme Test
DSP	Digital Signal Processing
LPC	Linear Predictive Coding
MBROLA	Multi-Band Re-synthesis Overlap-Add
MOS	Mean Opinion Score
MRT	Modified Rhyme Test
NLP	Natural Language Process
NSW	Non Standard Word
PCM	Pulse Code Modulation
RELPC	Residual Excited Linear Predictive Coding
TCL	Tool Command Language
TD-PSOLA	Time Domain Pitch Synchronous Overlap Add
TTS	Text To Speech

Chapter One

Introduction

1.1. Background

The advent of the information age places increasing demand on technologies to provide universal access. For information to be truly accessible, especially to the technology naive, anytime anywhere, one must seriously address the problem of user interfaces. A solution to this problem is to impart human like capabilities onto machines, so that they can speak and hear, just like the users with whom they need to interact (Lemmetty, 1999).

A technology that enabled the human-computer dialogues are now evolving for different languages such as English, French, Italy, etc. Such spoken dialogue systems according to Mc Tear (2002), involve the integration of a number of technologies that typically provide the following functions.

One of the functions is Speech Recognition. This is responsible for the conversion of an input speech utterance, consisting of sequence of acoustic-phonetic parameters, into a string of words. The other one is, Language Understanding, which attempts to analyze string of words with the aim of producing a meaning representation for the recognized utterance that can be used by the dialog management.

The third function that controls the interaction between the system and the user, including the co-ordination of the other components of the system is called a Dialog Management. There is also a User Interface that enables communication with external system, response generation and speech output. This Communication with External System enables to easily interact, for example with a database system, expert system, or other computer applications; whereas Response generation is responsible for the specification of the message to be output by the system. Finally, a speech output module uses a text-to-speech synthesis or pre-recorded speech to output the system's message.

These components, by interacting with each other, can retrieve the requested information from the external source, and construct the message that is to be sent to the speech output component to be taken to the user, through the response generator.

Speech is the usual and common mode of communication between human beings. When a person hears speech in his/her own language, the person hears the individual words and sounds. This can easily be proven in that, speech can be written using discrete letters and spaces between words. But this is not true if the person hearing the speech is not familiar with the language. In this case all the words and sounds seem to run together in a continuous stream (Rodman, 1999).

Text-To-Speech (TTS) system is a process which artificially produces synthetic speech for various applications such as services over telephone, e-document reading, and speaking system for handicapped people etc. (Swee, 2004); and Synthesized speeches can be created by concatenating pieces of stored speeches on the database. A computer based system that should be able to read any text audibly, whether it was directly introduced in the computer by an operator or scanned and submitted to an Optical Character Recognition (OCR) system is termed as *TTS synthesizer* (Dutoit, 1996).

There can be various synthesizers developed today, but, the quality of a speech synthesizer is judged by its similarity to the human voice or naturalness and by its ability to be understood which we call intelligibility (Klatt, 1987).

We can then broadly classify the existing Speech Synthesis systems into natural and artificial speech synthesis systems, where the natural synthesis is the one human being is using to communicate with each other, and, artificial synthesis is the one that computers use to communicate with human.

Although machines that imitate the voice of human are far from reality, computers with the power to speak and listen are enough to create a friendly, eyes free environment for the user at this time. For this reason, human-computer or machine communication via speech has become a shared vision and common goal of many speech researchers (Morka, 2001).

Building a speech synthesis system, a system that can probably acts on behalf of a human being, had been one of the areas that many researchers wanted to make it real. Today, we can easily see the human-computer interaction and its upcoming development evolving.

According to Syrdal et al (1998), the three major areas that should be addressed to improve the currently available speech synthesis technology in terms of naturalness are: improved linguistic analysis that mainly deals with syntactic parsing, assigning word pronunciation and identifying lexical stress. The second one is, improved prosody modeling, which mainly focuses on conveying both linguistic and extra-linguistic information about a speaker's attitude, intentions and physical or emotional state and the last one is improved speech synthesis model that depends on the above two models mentioned.

1.2 Speech Technologies

Though there are a number of speech technologies, the mechanisms, the techniques and tools used may vary based on the type and function of their existence. But the common function of these speech technologies is producing synthetic speech wave forms. According to Black and Lenzo (2000), there are three methods to generate synthesized speeches: *Articulatory, Formant and Concatenative.*

Articulatory synthesis is one of the methods where the human vocal tract, tongue, lips, etc are modeled in a computer. This method can produce very natural sounding samples of speech but at present it requires too much computational power to be used in pratical systems.

Further decomposition of the speech signal allowed the development of *formant synthesis*, where collections of signals were composed to form recognizable speech. The prediction of parameters that compactly represent the signal, without the loss of any information critical for reconstruction, has always been, and still is, difficult. Early versions of formant synthesis allowed these to be specified by hand, with automatic modeling as a goal. Today, formant synthesizers can produce high quality, recognizable speech if the parameters are properly adjusted, and these systems can work very well for some applications. It's still hard to get fully natural sounding speech from these when the process is fully automatic – as it is from all synthesis methods.

Formant synthesis does not use any human speech samples at runtime (Schröder, 2005). Instead, the output synthesized speech is created using an acoustic model. Parameters such as fundamental frequency, voicing, and noise levels are varied over time to create a waveform of artificial speech. This method is sometimes called Rule-based synthesis (Assaf,

2005). But some argue that because many concatenative systems use rule-based components for some parts of the system, like the front end, the term is not specific enough.

Concatenative synthesis is one of the simplest methods for speech synthesis and at the same time bypasses most of the problems encountered by articulatory and formant synthesis techniques mentioned so far, for detail discussion refer Section 2.4.3.

For this research work, concatenative based synthesis is used because of the above benefits of the technology; and many systems that based on formant synthesis technology generate artificial and robotic-sounding speech, and the output would never be mistaken for the speech of a real human. However, maximum naturalness is not always the goal of a speech synthesis system (Black and Lenzo, 2000).

1.3 Statement of the Problem

As to Assefa (2005) and the Ethiopian Statistical Agency' report of 2007 indicates, the total number of Afaan Oromo speakers is estimated to be more than 30 million, ranging from Ethiopia to Tanzania. Furthermore, Afaan Oromo language is used as a medium of instruction in the primary schools in the Oromiya region and as an official language in the region and some regional offices in Addis Ababa. The existence of a huge amount of printed

materials specific to the language has also arisen the need of such intelligent systems, which otherwise would be difficult to manage documents for reader`s.

In the context of the Afaan Oromo Language, the lack of a proper TTS system for the language's speakers can be raised as the other main problem behind conducting the research. As to Zegaye (2003), the systems that have been developed so far are meant to serve a specific language and totally limited to some 'techno-rich' countries of the world. In addition, a mere acceptance of one's language speech synthesizer will not help to work for others. This language dependency nature of speech synthesizer needs decisive investigations, explorations and researching in general, for a particular language.

The wide application areas mentioned above have made TTS system to be developed for many local languages: Morka (2001), Henock (2003) Tesfaye (2004) and Tewodros (2009) tried to develop TTS system for Afaan Oromo, Amharic, Tigrigna and Wolaytta languages, respectively; while Alula (2010) tried to build Amharic TTS system for Non Standard Word's (NSW's). Indeed, this has initiated the researcher to further extend the research works of Morka (2001).

Though Morka attempted to build a TTS system at word level using diphones as speech units, the considerations of prosodic effects and speech synthesis techniques were not applied on the developed system. The prosody of a speech or the way things are spoken is

an important part of speech message (Black and Lenzo, 2000) there for, changing the placement of emphasis in a sentence can change the meaning of a word, and this emphasis might be revealed as a change in pitch, volume, voice quality, or timing.

So far, the higher level phones specifically, the Triphones[1], had not been considered for any researches done for local languages as to the researchers knowledge; which higher level phones in the structure of Afaan Oromo language plays crucial role to control the prosody effects in speech synthesis.

For this study, the researcher tried to control the prosody effects so that the speeches that are to be synthesized should resemble the way a human utters. This is handled using the speech synthesis technique called Residual Excited Linear Predictive Encoding (RELPC) for the cases that, RELPC is used to perform pitch and duration modifications directly on continuous waveforms without using any parametric model as stated by Dutoit (1997).

Therefore, the above facts can show us that, it is compelling to extend the works done by Morka (2001) and study the TTS system for the language considering the prosodic effects and Triphones speech units.

[1] Triphones are phone units that are a combinations of three phones

1.4 Objective

1.4.1. General Objective

The main objective of this research is to explore the possibility of developing a concatenative speech synthesizer for Afaan Oromo language and test the functionality of the system on selected datasets to see the extent of performance improvement, intelligibility and naturalness.

1.4.2. Specific Objectives

In order to meet the general objectives mentioned above, the following specific objectives are set**:**

- To review different literatures on Natural Language Processing and Digital Signal Processing (DSP) techniques so as to see how languages are processed and speeches are rendered automatically from the database constructed;
- To review related literatures on phonology of Afaan Oromo and works done on speech synthesis in general, so that detail knowledge of the structure of the language can be grasped and to identify which parts of the speech techniques and technologies are discovered.

- To organize a list of Afaan Oromo language phones from different literatures reviewed on phonology of the language;
- To explore the available TTS tools and algorithms in order to select more convenient one for this specific research work;
- To design a Text To Speech synthesizer prototype for Afaan Oromo language using diphone and triphone units;
- To evaluate the performance of the system using test data sets selected from Afaan Oromo texts;
- To provide concluding remarks and recommendations for further research works in this area.

1.5 Scope and Limitations of the Study

Developing a speech synthesis system for Afaan Oromo Language at phrase level is the main area of focus for this work; with prosody rules into considerations.

In this specific work, utterances are speaker dependent and the diverse regional dialects of the Afaan Oromo language are not taken into considerations.

Due to lack of ready-made corpus data for the language, this work is done to try to show the possibility of developing a prototype for Afaan Oromo language, and hence, a limited

number of words and sentences are used to build a corpus for experimentation though this indeed has got a problem of uttering speech units that are new for the system.

In this research, the speech units are limited to diphone and triphone; the lower level speech units like monophones and the next higher level units such as quadrophones are not the interests of this work to explore.

Due to a need of detail explorations of the festival system, phrases that included the special character for the Afaan Oromo language, which is “’” (hudha), are not included in the data set.

1.6 Methodology

The methodology part of this research mainly focuses on the ways and procedures used to conduct this research. To undertake this specific research the following methodologies are followed.

1.6.1 Literature Review

The investigations of the theoretical concepts are given a high emphasis to start this research work from the very beginning. Various studies in literatures like Books, Journals,

Conference proceedings and the Internet are reviewed with regard to the language; specifically, literatures in the areas of Natural Language Processing (NLP) including units of a language and Digital Signal Processing (DSP) are the focus of extensive review points. In addition, TTS rules and techniques are also investigated.

1.6.2 Program Development Tool

To develop a TTS prototype for Afaan Oromo, a Festival Text To Speech tool is used. Festival, which as to Black and Lenzo (2000) is an open-source, stable and portable multilingual speech synthesis framework developed at the Center for Speech Technology Research (CSTR), of the University of Edinburgh. It provides a basic utterance structure, a language to manipulate it, and methods for construction and deletion. It also interacts with audio system in an efficient way, spooling audio files while the rest of the synthesis process can continue.

To generate the diphones and triphones Cygwin which is also an open source Linux based for windows software has been tried. Though it has the feature to be embedded into windows plateform, generating the triphones were not as such an easy task to be handled than working in festival.

Hence, for it is a recent technology, ease for access, used for the speech synthesis and the familiarity of the researcher with the tool and the algorithms it is built with, Festival system is preferred for this work.

1.6.3 Testing and Evaluations Techniques

Depending on what kind of information is needed, the evaluations of any TTS system can be made at phoneme, word, or sentence level. There are many tests that help to address these three issues. But basically, the overall TTS system is assessed based on the Intelligence and Naturalness (Mouran, *n.d*). The *Intelligibility* of a system is how much of the spoken output the user understands, as well as how quickly a listener gets fatigue by only listening, where as the *Naturalness* of a system is the measure of how much like real speech does the output of the TTS system sound.

The Mean Opinion Score (MOS) is the commonly used techniques to evaluate the overall performance of any TTS system even its Naturalness. MOS uses scale level that ranges from 1 (bad) to 5 (Excellent) as per to the systems performance. As it is stated by Lemmetty (1999), to use this technique, native speakers of the language are invited to listen and respond to the accuracy of the uttered words, while the system reads the given input text.

For this specific research work, the Mean Opinion Score (MOS) scale is used in preference to the others for the case that it is widely used and the simplest method to evaluate the overall performance of a speech quality.

1.7 Significance of the study

Conducting this speech synthesis research for Afaan Oromo language has tremendous advantage apart from the academic fulfillment of the thesis work.

This research will be a strong asset for a research conducted on Human Computer Interaction in general, and its outcome will be one of the doors for the development of a full fledge Speech Synthesis for Afaan Oromo language.

Since the Afaan Oromo language is a vast one with huge amount of speakers, one can figure out easily where the developed system's area of application is. Probably the most important and useful application field in speech synthesis is the reading and communication aids for the blind and aid for the Deafened and Vocally Handicapped. People who are born deaf can't learn to speak properly and people with hearing difficulties have usually speaking difficulties. This Synthesizer gives the deafened and vocally handicapped an opportunity to communicate with people who do not understand the sign language. The significance of this TTS system for Afaan Oromo language can be coupled with a Computer Aided

Learning system, and provide tools to help a child or a student learn correct pronunciation of words.

Since the newest applications in speech synthesis are in the area of multimedia, and some systems that can read electronic mail were developed so that a customer uses his telephone to read his emails (Eker, 2002), this TTS system for Afaan Oromo language can be a base line for the development of multimedia oriented TTS system.

1.8 Organizations of the Research Work

This research report is organized into six chapters where each chapter explicitly tells what have been performed at each stage to come up with the output.

The first chapter is the introductory part of the speech synthesis under investigation, the problem statement, the objectives, the methodology, the significance of the study and finally the scope and limitations of the work undertaken.

The second chapter is mainly the explorations of the literature reviews on human speech system, basics of TTS system, types of speech synthesis, and finally the overview of research works done by foreign and local researchers on the same field under investigation.

The rules and writing system of Afaan Oromo language with the overall phonology of the language are explored in chapter three of this work. The chapter starts with the language

itself and continues with the components of the writing systems and finally, discussions are made on the phonology of the language.

Chapter four mainly deals with the current Text To Speech system algorithms: Linear Predictive Coding, TD-PSOLA and RELPC. Finally, it proposes the selected algorithm, RELPC, and explores its advantage over the others.

The fifth chapter is all about the experimentations and evaluations of the Afaan Oromo language TTS system prototype built.

The last chapter, chapter six, is the conclusions and recommendations based on the experiment and findings of this research to show an opening track for future research conducted on TTS system.

Chapter Two

Literature Review

The synthesis of a speech can be seen broadly from two major perspectives (Moutran, *n.d.*): Natural point of view and the Artificial. The Natural speech synthesis is what humans do while communicating and the speech that we call Artificial can be synthesized in a number of ways (articulatory, formant or concatenative). But both synthesized speeches incorporate the knowledge of the target language; while artificial synthesis needs technical knowledge like signal processing in addition.

This chapter starts with the discussions of the Human Speech System and tries to touch basics of speech synthesis components, synthesis units and synthesis methods.

2.1 The Human Speech System

The idea of giving computers the ability to process human language is as old as the idea of computers themselves (Jurafsky and Martin, 1999).

The main components of the speech production organizations are Lungs, Trachea (wind pipe), Larynx, Pharyngeal oral or Buccal cavity (mouth), and Nasal cavity (nose) as it is discussed by Eker (2002). The pharyngeal and oral cavities are usually grouped into one unit referred to as the vocal tract and the nasal cavity is often called the nasal tract.

Accordingly, the vocal tract begins at the output of the larynx (vocal cords, or glottis) and terminates at the inputs to the lips. The nasal tract begins at the velum and ends at the nostril. When the velum (the trapdoor like mechanism at the back of the oral cavity) is lowered, the nasal tract is acoustically coupled to the vocal tract to produce the nasal sounds of speech. The pharynx is the tube like organ extending from the back of the mouth to the larynx.

The vocal tract is bounded by hard and soft tissue structure. These structures are either essentially immobile (such as hard palate and teeth), or are movable. The movable structures associates with speech production are also referred to as articulated. The tongue, lips, jaws, and velum are the primary articulators; movements of these articulators appear to account for most of the variations in vocal tract shape associated with speaking. However, additional structures are capable of motion as well; for instance, shorten or lengthen the vocal tract (Wouters, 1996).

A speech production model that more directly simulates the physical process of human speech production comprises lungs, vocal cords, and the vocal tract. The vocal cords are expressed as a simple vibration model, and the pitch of the speech changes according to adjustments in the tension of the vocal cords. When the vocal cords are closed, their vibration results in voiced sounds; when they are opened, this vibration stops, and

unvoiced sounds result. The vocal tract model is created as a non-uniform sound tube with differing cross-sectional areas, and the transmission of sound waves inside the sound tube is expressed by a digital filter. Vocal tract resonance characteristics are controlled according to the cross-sectional area (vocal tract area function) for various parts of the vocal tract.

The speech production process has many levels, from the movement of vocal organs to the production of sounds. There are four hierarchical levels in speech production: the speech sound level, vocal tract shape level, vocal organ configuration level, and muscle contraction level. There is a "one-to-many" relationship among levels starting from the sound level and moving toward the muscle contraction level. For example, as is evident in the case of ventriloquism[2], sounds that seem very similar can be created using different vocal tract shapes. Similar vocal tract shapes can be created using differing vocal organ configurations; for example, the degree of mouth opening is determined by the relative position of both lips and the jaw. Furthermore, each individual vocal organ involves two or more competing muscles, and the vocal organ configuration is determined by their relative degree of contraction. This means that when speech with a given tone is created, the "one-to-many" relationship that exists among levels cannot be determined uniquely because there is always an excessive degree of freedom on the lower level (Honda, 2003).

[2] Skill of speaking without moving the lips

To produce sounds, the oral tract must involve an active articulator, which is raised to form the stricture, as well as a passive articulator towards which the active articulator is raised. Table 2.1 is all about the major places of Active and Passive articulators

Place	Active Articulator	Passive Articulator
Bilabial	Lower Lips	Upper Lips
Labio-Dental	Lower Lips	Upper Teeth
Dental	Tip of Tongue	Upper Teeth
Alveolar	Blade of Tongue	Alveolar Ridge
Retroflex	Tip of Tongue	Hard Palate
Palatal	Front of Tongue	Hard Palate
Velar	Middle of Tongue	Velum (Soft Palate)
Uvular	Back of Tongue	Uvula

Table 2.1 Major Places of Articulations

(Adopted from Mongham, *n.d.*)

The number of places along the oral tract where a stricture can be produced is the theoretically infinite (same as “how many points are there on a line?"), but human languages only distinguish a dozen or so and most language only use about half of those (Mongham, n.d)

The main energy source is the lungs with the diaphragm. As Figure 2.1, the structure of human vocal organ, cleary depicts, when speaking, the air flow is forced through the glottis between the vocal cords and the larynx to the three main cavities of the vocal tract; the pharynx, the oral and nasal cavities. From the oral and nasal cavities the air flow exits

through the nose and mouth, respectively. The V-shaped opening between the vocal cords, called the glottis, is the most important sound source in the vocal system. The vocal cords may act in several different ways during speech. The most important function is to modulate the air flow by rapidly opening and closing, causing buzzing sound from which vowels and voiced consonants are produced. The fundamental frequency of vibration depends on the mass and tension and is about 110 Hz, 200 Hz, and 300 Hz with men, women, and children, respectively (Black and Lenzo, 2000). With stop consonants the vocal cords may act suddenly from a completely closed position, in which they cut the air flow completely, to totally open position producing a light cough or a glottal stop. On the other hand, with unvoiced consonants, such as /s/ or /f/, they may be completely open. An intermediate position may also occur with for example phonemes like /h/.

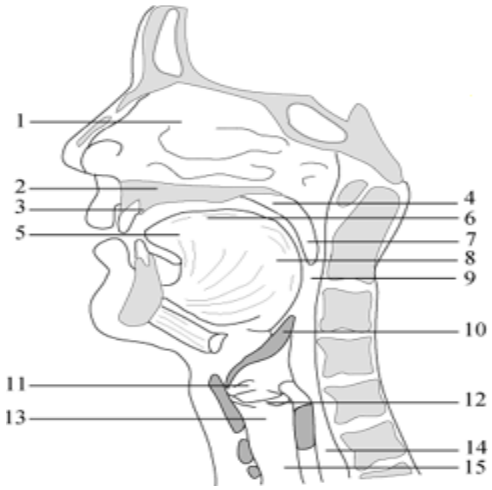

Figure 2.1 The human vocal organs[3].

(Adopted from Rabiner, 1993)

The pharynx connects the larynx to the oral cavity. It has almost fixed dimensions, but its length may be changed slightly by raising or lowering the larynx at one end and the soft palate at the other end. The soft palate also isolates or connects the route from the nasal cavity to the pharynx. At the bottom of the pharynx are the epiglottis and false vocal cords to prevent food reaching the larynx and to isolate the esophagus acoustically from the vocal tract. The epiglottis, the false vocal cords and the vocal cords are closed during swallowing and open during normal breathing.

[3] (1) Nasal cavity, (2) Hard palate, (3) Alveolar ridge, (4) Soft palate (Velum), (5) Tip of the tongue (Apex), (6) Dorsum, (7) Uvula, (8) Radix, (9) Pharynx, (10) Epiglottis, (11) False vocal cords, (12) Vocal cords, (13) Larynx, (14) Esophagus, and (15) Trachea.

2.2 The basics of Text-To-Speech system

The total sum of the idea behind the development of any Text-To-Speech revolves around the concept of artificially synthesized speech which indeed, practically involves two major tasks. The first one is a Natural Language Processing (NLP) which is a high level synthesis where an input text is transcribed into a phonetic equivalent or some other linguistic representation (Lemmetty, 1999).

The second phase which is the speech synthesis phase is a phase where speech waveforms are produced from the phonetic data of the first phase and some prosodic information. This phase is also known as low-level synthesis. Figure 2.2 is a generalized pictorial representation of the two phases of TTS system (Dutoit, 1997):

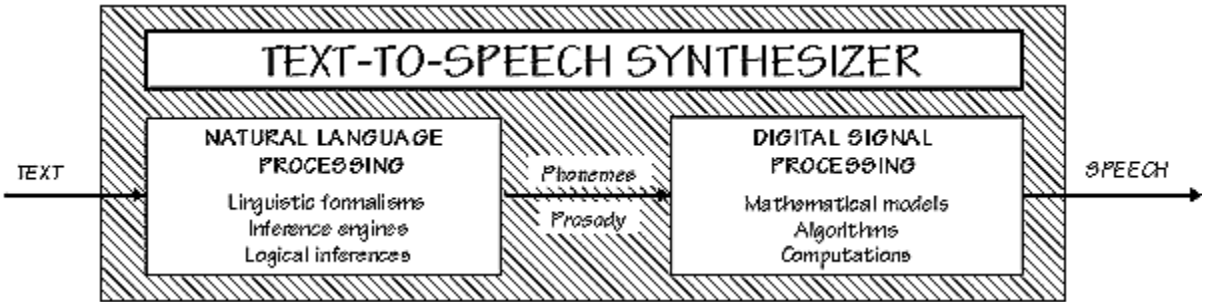

Figure 2.2 A generalized TTS System Architecture

(Adopted from Dutoit, 1997)

Any Text To Speech system contains components performing the above two phases. The high level synthesis is carried out by the Natural Language Processing (NLP) component of the TTS system where as, the low level synthesis on the other hand, is carried out by the Digital Signal Processing (DSP) component of the TTS system (Lemmetty, 1999).

2.2.1 The Natural Language Process Component

This part of the process is often the complex one of the two components. Numerals, abbreviations, and acronyms all need some preprocessing techniques to convert them into the corresponding full words. Correct prosody and pronunciation analysis from written text is also a difficult task because written text does not contain explicit emotions that are expressed while speaking (Lemmetty, 1999).

The task of converting the input text into a linguistic representation can be further partitioned into two components: the transformation of text into **phonetic units** and the conversion of text into **prosodic parameters** (Ng, 1998).

2.2.1.1 Phonetic Units

- ✓ **Text Normalization**

Unrestricted texts include Standard Words (common words and Proper Names) and Non-Standard Words (NSWs). Standard Words have a specific pronunciation that can be

phonetically described either in a lexicon, using a disambiguation processing to some extent, or by letter-to-sound rules. In the context of TTS the problem is to decide how an automatic system should pronounce a token; even before the pronunciation of a token, the Word Pronunciation is important to identify the NSW-Category of a token (Raj et al, 2007)

- ✓ **Text to phonetic units**

This is also called Phonetic Analysis which refers to the conversion of the linguistic description in orthographic form to phonemes. This component is responsible for grapheme-to-phoneme conversion for targeted language after the words are normalized. The input is the text structure from the previous component and the pronunciations for the words normalized are appended to this structure (Hasim, 2000).

2.2.1.2 Prosodic parameters

Prosody refers to variations in the pitch, duration and stress of speech. Nobody speaks without varying pitch under normal circumstances. In general, prosody gives additional information that the uttered words cannot give alone. In addition to conveying emotional or other connotations, it has also a changing effect on the meaning of the sentence (Rodman, 1999).

As it is described by Minghui (2000), prosody can be seen at different levels. At linguistic level, we know the tone, intonation stress of a speech. That is why prosody exhibits in linguistic level. It is also perceived by human as pitch, loudness, length and strength. This is what we get at perceptual level. Prosody, if expressed at acoustic level, is actually fundamental frequency, duration, amplitude etc…and that is what we can operate on the synthesis process. Indeed, the proper combination of these acoustic factors makes speech natural, expressive and active. The following can be mentioned as factors for contributing to the prosodic features (Lemmety, 1999):

- **Feelings** (Anger, Happiness, Sadness)
- **Speaker Characteristics** (Gender, Age, Dialects)
- **The meaning of the Sentence** (Neutral, Imperative, Question)
- **Fundamental Frequency** (Duration and Stress)

Unfortunately, written text usually contains very little information of the above mentioned features of the prosody and some of them change dynamically during speech. However, with some specific control characters this information may be given to a speech synthesizer; so that the synthesizer resembles the speeches of a human being.

2.2.2 The Digital Signal Processing

After the input texts are pre-processed, the next step will be synthesizing speech utterances based on the outcomes of the text analysis. The generation of speech waveforms is the production of acoustic output from phonetic and prosodic information (Lemmety, 1999). The DSP module takes in the output of the NLP module (i.e. phonetic transcription and prosodic information) and produces a corresponding speech signals using one of the three known methods: Articulatory, Formant or Concatenative synthesis methods.

2.3 Types of Speech synthesis

Speech signal generators (the *synthesizers*) can be classified into three categories: Articulatory Synthesizers, Formant Synthesizers and Concatenative Synthesizers.

2.3.1 Articulatory Synthesizer

Articulatory synthesizers are physical models based on the detailed description of the physiology of speech production and on the physics of sound generation in the vocal apparatus. Typical parameters are the position and kinematics of articulators. Then the sound radiated at the mouth is computed according to equations of physics. Experiments with articulatory synthesis systems have not been as successful as with other synthesis systems but in theory it has the best potential for high-quality synthetic speech. For

example, it is impossible to use articulatory synthesis for producing sounds that humans cannot produce (due to human physiology). In other synthesis methods, it is possible to produce such sounds and the problem is that, these sounds are usually perceived as undesired side effects. The articulatory model also enables more accurate transient sounds than other synthesis techniques (Lemmetty, 1999). Because of the reasons mentioned above and its cost in terms of computation and the underlying theoretical and practical problems still unsolved, this type of synthesizer is rather far from applications and marketing

2.3.2 Formant Synthesizer

Formant synthesis is a descriptive acoustic-phonetic approach to synthesis. Speech generation is not performed by solving equations of physics in the vocal apparatus unlike that of Articulatory, but rather by modeling the main acoustic features of the speech signal. The basic acoustic model is the source/filter model. The filter, described by a small set of *formants*, represents *articulation* in speech. It models speech spectra that are representative of the position and movements of articulators. The source represents *phonation*. It models the glottal flow or noise excitation signals. Both source and filter are controlled by a set of phonetic rules (typically several hundred). This, of course, is interesting for modeling emotional expressivity in speech and has been a strong asset for high-quality rule-based formant synthesizers, including multilingual systems, to the market for many years.

However, for the cases that no human speech recordings are involved at run time, the resulting speech sounds relatively unnatural and "robot-like". This, of course, is interesting for modeling emotional expressivity in speech (Schröder, 2005).

2.3.3 Concatenative Synthesizers

Concatenative synthesis is based on speech signal processing of natural speech databases. It operates, as it is indicated by Minghui (2000), in such a manner that, appropriate speech units are concatenated to construct the required speech. The segmental database is built to reflect the major phonological features of a language. For instance, its set of phonemes is described in terms of diphone units, representing the phoneme-to-phoneme junctures. Non-uniform units are also used (diphones, syllables, words, etc.). The synthesizer concatenates (coded) speech segments, and performs some signal processing to smooth unit transitions and to match predefined prosodic schemes. Direct pitch synchronous waveform processing is one of the most simple and popular synthesis algorithms. Other systems are based on multi-pulse linear prediction, or harmonic plus noise models.

Concatenative synthesis is probably the easiest way to produce intelligible and natural sounding synthetic speech as described by Lemmety (1999). But, despite its easiness, concatenative synthesis is usually limited to only one speaker and one voice; moreover, the method often requires more memory capacity than other methods.

Formant synthesizers may sound smoother than concatenative synthesizers because they do not suffer from the distortion encountered at the concatenation point. To reduce this distortion according to Huang et al. (1996), concatenative synthesizers often select their units from carrier sentences, or monotone speech, and/or perform spectral smoothing, all of which can lead to a decrease of naturalness. The resulting synthetic speech may not resemble the donor speaker in the training database. Therefore, another data-driven approach used to minimize the number of concatenation points so as to reduce the distortions is to select larger units, such as triphones or words.

But, according to Lemmety (1999), one of the most important aspects in concatenative synthesis is, to find correct unit length. The selection is usually a trade-off between longer and shorter units. With longer unit high naturalness, less concatenation points and good control of co-articulations are achieved, but the amount of units and memory required is increased. With shorter units, less memory is needed, but the sample collecting and labeling procedures become more difficult and complex.

During concatenation, Phonemes, Diphones, Triphones, Syllables, and Words are the basic speech units that are in use these days and will be explained as follows.

Phonemes are the minimal distinctive phonetic units and relatively small in number in any targeted language but their main problems are they ignore transitional sound and need many concatenation points.

Syllable units are smaller than words and larger than a phoneme that usually consists of 2-3 phones. The number of different syllables in each language is considerably smaller than the number of words, but the size of the database is still larger for TTS system. But unlike words, co-articulation between syllable units may not be so weak, and as a result smoothening across unit boundaries using the available algorithms like TD-PSOLA and Residual LPC will not be as such easy (Donovan, 1996).

Diphones are made up of two phonemes to incorporate transitional sound for a better sounding speech. A diphone extends from the central steady part of one phone to the central part of the next phone. This enables the concatenation of units in the steadiest region of the signal, and consequently reduces the distortion from concatenation points (Donovan, 1996).

Triphones are also one of the speech units resulted from any combinations of three phones for a better transitions between utterances. The triphone unit is the basic phone model in many current phonetic speech technologies. The reason for this is that triphones capture the coarticulation effect caused by the immediate preceding or following phonetic context

though their main drawback is that the triphone inventory is quite large Blomberg and Elenius (*n.d.*).

A word concatenation is the most natural unit for written text and some messaging system with limited vocabulary such as Airplane reservation and weather forecast reports. The concatenation of words is relatively easy to perform and co-articulation effects within a word are captured in the stored units (Vosnidis, 2001). However, employing such long units is not practical from the point of view of TTS synthesis, where an arbitrary text could appear at the input of the system, i.e. the TTS sytem should make rooms for any word to be synthesized (not only the given words).

In addition, for the cases that a word spoken in isolation and a word spoken in a continuous sentence are totally different, as to Donovan (1996), it makes the result a little bit unnatural and difficult to understand because of the pitch and formant discontinuities at the boundaries of words.

The most common choices for speech synthesis are phonemes and diphones because they are short enough to attain sufficient flexibility and to keep the memory requirements reasonable (Swee, 2004). Specially, the use of diphone in the concatenation provides rather good possibilities to take account of coarticulation because it contains the transition from

one phoneme to another; the latter half of the first phoneme and the former half of the latter phoneme.

Earlier, using longer units such as syllables or words were impossible or impractical for reasons that they require huge amount of memory. But nowadays, memory is not a matter of question; in addition to diphones, triphones are also becoming the state-of-the-art to represent the utterance of a speech in a language at times diphones are not sufficed. Therefore, for this research these two units' diphones and triphones are the area of focus.

2.4 The Festival System Architecture

The Festival Speech Synthesis System was developed at the Center for Speech Technology Research at the University of Edinburgh in the late 1990s by Alan Black, Paul Taylor and Richard Caley. It is designed for three particular users of speech synthesis: speech synthesis researchers, speech application developers, and the end user (Black and Lenzo, 2003).

Festival is also designed to allow the addition of new modules in an easy and efficient way as it is discussed by Black and Lenzo (2003). Another aspect which makes the Festival system more compatible is that Festival is not simply used for researching into new synthesis techniques, but also used as a platform for developing and testing Text-to-Speech systems as well as a fully usable Text-to-Speech system. This is good for embedding into other projects that require speech output.

Festival is implemented in two languages, C++ and scheme (a variant of lisp). While in principle it would be attractive to implement the system in a single language, practical reasons concerning the nature of programming languages necessitate the approach it took. Scheme has a very simple syntax according to Black and Lenzo (2003); but it is at the same time powerful for specifying parameters and simple functions. Scheme is chosen because it is restricted and is considered as a small language and would not increase the size of the Festival system.

In addition to being a research platform, Festival operates as a run-time system and hence speed is vital.

Festival is also a general text-to-speech system that uses the text analysis schem (Tokenization and Normalization), phonetic analysis (Grapheme to phoneme conversion), prosodic analysis (pitch and Duration Attachement) and Residual-LPC (Linear Predictive Coding) synthesis technique, which is able to transcribe unrestricted text-to-speech.

The Normalization process of a text is one of the processes of text analysis. After the tokenization process, the tokenizer handed-over the tokenized words to the normalizer. The goal of the normalizer is to transform the raw input text stream into a regularized format that can be processed by the rest of the system. This includes expanding numbers, dealing with abbreviations and possibly tagging words with their part of speech labels to help with later pronunciation and prosody processing (Ng, 1998). For example *"ganda 1 qabu"* ("they have 1 Village") after tokenization it should have been normalized as *"ganda"*,*"tokko"* and *"qabu"*;

where the normalization effect is seen in "1" to be normalized as *"tokko"*. The token-to-word rule that is supported by festival enables the normalization of tokens to their specific utterances in the database built, but, it is language dependent.

With LPC method, the residuals and LPC coefficients are used as control parameters. LPC methods are the most widely used in speech coding, speech synthesis, speaker recognition and verification and for speech storage.

LPC methods in addition provide extremely accurate estimates of speech parameters, and do it extremely efficiently (Black and Lenzo, 2003).

2.5 Testing Mechanisms

There are different testing mechanisms used for speech synthesis technologies; but the common and the most frequently used are detailed as follow:

Diagnostic Rhyme Test (DRT), according to Sak (2000), is a test of the intelligibility of word-initial consonants. Subjects are played pairs of words with a different first consonant, and asked to identify which word they heard (e.g., thing vs. sing). Six contrasts are represented, namely: voicing, nasality, sustention, sibilating, graveness, and compactness. The system that performs best is the one that produces the lowest error rate. Usually, only total error

rate percentage is given, but also single consonants and how they are confused with each other can be investigated.

This is a very effective method and can be used since it is an easy and reliable method. However, for this method does not test any vowels or prosodic features, the technique is not used to assess the overall performance of Afaan Oromo language TTS system.

The other one, as it is described by Lemmety (1999), is Modified Rhyme Test (MRT). This is one of the evaluation techniques, where evaluation is done to assess the system's intelligence. MRT is like DRT, but includes tests both for word-initial and word-final intelligibility (e.g., bath vs. bass). The first half of the words is used for the evaluation of the initial consonants and the second one for the final ones. Results are summarized as in DRT, but both final and initial error rates are given individually. As to Moutran (*n.d.*), in this kind of test, evaluators are asked to play the system and compare it to other systems. The same set of sentences is used and ranks are given as to the renditions of each sentence.

But, for the reasons that comparing the Afaan Oromo TTS system with other systems is not the scope of this research work, we don't use this technique for this specific work.

The third type of testing technique is a Mean Opinion Score (MOS) scale. This is also one of the commonly used techniques as far as a speech technology is concerened and perhaps it is also used for this research work.

2.6 Related works done

Different literatures, that give more concern to the speech synthesis system, are reviewed. Though there are many related works, the following review is some of the summary of related works done on TTS system that had been conducted on local and foreign languages.

2.6.1 Related works done on Foreign Languages

Malay Text to Speech (Malay TTS) system developed by Swee (2004) is a diphone based system that uses a RELPC to smooth the errors occurred during utterance. The researcher designed a set of prosody rules using a CART tree as the preliminary study in prosody design for Malay TTS system. The result found after testing using the MRT test mechanism is 89.2% which is near to intelligent system.

Works done by Assaf (2005), which is entitled "A Prototype of an Arabic Diphone Speech Synthesizer in Festival", is built for Arabic language. After testing using MRT the researcher came across with 85% for word testing and 75% for sentences testing and finally

recommended writing syllable rules since Arabic relies more on stress rather than intonation; the researcher also recommended to control the prosody.

The other research work to be mentioned here is entitled with "Concatenative Synthesis of Persian Language Based on Word, Diphone and Triphone Databases". This is found to be the first Persian Concatenative TTS built by Javidan and Rasekh, (2010). Artificial neural network with unsupervised learning paradigm is used to build the system using different types of speech units. Moreover, the model is used to synthesize the desired utterances which are: words, diphones and triphones. The experimental results over the system showed its ability to produce unlimited number of words with high quality voice and high accuracy in converting the written text into speech. The resulting model obtained an accuracy of 99% for the word and diphone models and 86.5% for the triphones.

2.6.2 Related works done on Local languages

The speech synthesis system tried by Tesfay (2004), entitled with "Diphone Based Text-To-Speech System for Tigrigna Language" is developed using TD-PSOLA (Time-Domain Pitch Synchronous Overlap and Add) technique for diphone concatenation and MATLAB programming language to handle the signal processing. Testing was done based on the Mean Opinion Score scale level and the average result obtained was 3.05, which is closer to

the scale level "good". Then finally, a better text processor was recommended to increase the naturalness of the system.

The research entitled with a "Unit Selection Voice for Amharic Using Festvox" (Sebsibe et al 2004), is tried to address issues considered in developing a concatenative speech synthesizer for Amharic Language. They developed a unit selection concatenative speech synthesizer by using transliteration scheme. As it has been indicated in the research paper, Festvox, which is a voice building framework used for building unit selection voices in a new language is used. The perceptual evaluation indicated in the research used six levels ranging from Excellent (5) to Very Poor (0) and resulted 2.9. The researchers finally recommended an increase in the number of the entries to the set of the corpus used for a better quality.

The first Afaan Oromo TTS system is tried by Morka (2001) while conducting a research entitled with "Text-to-Speech System for Afaan Oromoo". The technique that he used for the research he conducted was concatenative speech synthesis where diphones were used as the basic concatenation units to synthesize sample Afaan Oromo words. The thesis mentioned the consideration and inaccessibility of Hidden Markov Toolkit and Center for Spoken Language Understanding tools for the speech synthesis. It was also indicated in the research paper that success on recognizing the utterance of the transcribed phonetic unit was 43.33% for native speakers. He finally recommended the incorporations of smoothening

techniques like linear predictive coding (LPC) to smooth the transition points of the diphones to make improvement on the errors that arise from segmentation.

A research work by Henok (2003), is dealt with the same technique with that of Morka for Amharic speech synthesizer, but Henok has used Time-Domain Pitch Synchronous Overlap and Add (TD-PSOLA) technique to generate the synthetic speech. In addition to this, he has also considered prosodic effects, like anger, happiness and emotions, into account which Morka didn't.

A "TTS synthesizer for Wolaytta language" is one of the recent work conducted by Tewodros (2009) for the first time. The TTS was based on the diphone speech units where RELPC was considered as a smoothing technique. Finally, the performance of the system is measured to be 78% where as the naturalness and the intelligence of the system as to the MOS scale levels were 2.77 and 3.17 respectively.

The other recent work entitled with "A generalized approach for Amharic TTS system" was done by Alula (2010) on the possibility of developing a generalized approach in combining SW's and NSW's for Amharic language. As to the findings, 73.35% was the performance of the system built where as 2.83 and 3 were the naturalness and the intelligence of the system respectively. Final recommendations to shift from rule based approach to statistical based

approach was forwarded to convert NSW's to their equivalent SW's so as to incorporate all NSW's of Amharic language.

2.7 Discussion

Based on the reviews made on the local languages, and as to the researcher's knowledge, there is a single work (Morka, 2001) that attempts to design TTS system for Afaan Oromo language. Therefore, the researcher tried to extend works done on local language to a higher speech level, Triphone specifically, and tried to incorporate Prosody rules to diphones and triphones speech units.

Chapter Three

The Afaan Oromo Language

The Oromo are one of the major linguistic groups in Ethiopia. They live over a large area stretching from close to the Sudan border in the West, through Addis Ababa, and beyond Harar in the East, from Northern Kenya in the South, up and East of the rift valley, and to Wallo in the North (Lloret**,** 1997).

So far, Afaan Oromo is a single common mother tongue for the Oromo people. The language, Afaan Oromo belongs to the Eastern Cushetic group of language and is the most extensive of the forty or so Cushitic languages. It is very closely related to Konso, with more than 50% of the words in common, closely related to Somali and distantly related to Afar and Saho (Melba, 1988). It is considered to be also one of the five[4] most widely spoken languages from among the approximately 1000 languages of Africa (Lloret**,** 1997).

3.1 Writing System

The writing system used in Afaan Oromo is the Latin Qubee. The Latin Qubee has officially replaced the former writing system, Ethiopic Syllabary, of the language in 1991 (Tilahun,

1992). Listing in Table 3.1 shows Latin Qubee used in Afaan Oromo along with their pronunciations (Roba and Owens, 2003).

A a	B b	C c	CH ch	D d	DH dh	E e	F f	G g	H h	I i
[a]	[b]	[ʧʼ]	[ʧ]	[d]	[ɗ]	[e]	[f]	[g]	[h]	[i]
J j	K k	L l	M m	N n	NY ny	O o	P p	PH ph	Q q	R r
[ʤ]	[k]	[l]	[m]	[n]	[ɲ]	[o]	[p]	[pʼ]	[kʼ]	[r]
S s	SH sh	T t	U u	V v	W w	X x	Y y	Z z		
[s]	[ʃ]	[t]	[u]	[v]	[w]	[tʼ]	[j]	[z]		

Table 3.1 The Latin Qubee used in Afaan Oromo language with their Pronunciations (Adopted from Roba and Owens, 2003)

3.2 Phonology of Oromo

A phone is a speech sound; which is represented with phonetic symbols that bear some resemblance to a letter in an alphabetic language like English. So, for example, there is a phone represented by symbol *l* that usually corresponds to the letter *l* and a phone represented by *p* that usually corresponds to the letter *p* (Jurafsky and Martin, 1999). In some languages, some phones may be represented in two or more characters. For example, ch, dh, ny, ph, and sh are the commonly used single phones for the Afaan Oromo language.

Basically, phonology is the study of how sounds are organized and used in natural languages with the help of sounds, vowels and consonants.

[4] Kiswahili, Arabic, French and English

3.2.1 Consonants and Vowel

According to Jurafsky and Martin (1999), phones are divided into two main classes: consonants and vowels based on the motions of the air through the mouth, throat or nose.

3.2.1.1 Consonant

Consonants are made by restricting or blocking the airflow in some way, and may be voiced or unvoiced (Jurafsky and Martin, 1999).

Oromo has 24 native consonant phonemes; phonemes such as /p/, /Z/, and /V/ are shown reappeared in the language with loan words. Loan words which have /Z/ tend to be assimilated to /S/ (occasionally also to J), /P/ tends to be assimilated to /F/ (occasionally also as /P'/ or /V/) while /V/ tend to be assimilated as /F/ (Lloret, 1997)

e.g.

muuza, muusa, 'banana'

ajaja 'command' from Amharic azzaza

polisii, foolisii 'plice' from European origin

vitaamini, fitaamini, 'vitamin'

peesaa, beesee (southern) 'money' from Swhaili Pesa

According to Rodman (1999), three major factors determine the sound of a given consonant. These are:

- The place of articulation: the location of obstruction in the vocal tract.
- Manner of articulation: the relative position and activity of articulators while forming the obstruction.
- Voicing: the state of the vocal cords.

Afaan Oromo consonants occurs single in initial position while intervocally they may occur single, geminated, or as members of bi-consonantal clusters. In fact, consonant gemination is phonemic in Afaan Oromo (Tilahun, 1994).

Table 3.2 shows the Afaan Oromo Consonants. For the reason that in the language there are three consonants /p/, /v/ and /z/ that are loan words but are frequently used, this is shown explicitly with “’” mark in the table.

Consonants[5]					
	Labials	Alveolars	Palatals	Velars	Glottals
Plosives	B,(P)	T,D	ch /ʧ/ , j /ʤ/	K, G	' /ʔ/
Nasals	M	N	ny /ɲ/		
Ejectives	ph/p'/	x/T'/	c /ʧ'/	q/K'/	
Implosive		Dh/D/			
Fricative	F, (V)	S, (Z)	sh /ʃ/		H
Trill		R			
Lateral		L			
Glides	W	I	Y		

Table 3.2 Consonants in Afaan Oromo Language ('Dubbifama')

The Consonant /h/ in the Afaan Oromo language is the only consonant in the language that doesn't occur geminate (Lloret, 1997)

e.g

harree 'donkey' (double r) harka 'hand' (single r)

kale 'kidney' (single l) kalle 'child's garment' (double l)

butaa 'snatcher' (single t) butta 'period of Gada system' (doube t)

[5] All phonemes with ' indicates that they appear in loan words

3.2.1.2 Vowels

Vowels have less obstruction, and they are usually voiced, and are generally louder and longer-lasting than consonants. They are created whenever the air stream from the lung is unobstructed. Vowels can be classified by the position of the Tongue and Lips. The tongue and the lips produce different vowels by altering the shape of the vocal tract and enabling the vibrating air produce sounds in which different frequencies are emphasized (Rodman, 1999).

The Oromo language has five short basic vowels, two fronts, two back and one central which have all longer counterparts as it can be seen in Table 3.3.

	Front	Central	Back
High	i /ɪ/, ii /iː/		u /ʊ/, uu /uː/
Mid	e /ɛ/, ee /eː/		o /ɔ/, oo /oː/
Low		a /ʌ/	aa /ɑː/

Table 3.3 Vowels in Afaan Oromo Language ('Dubbachiiftu')

3.2.2 Morphological process

Morphology is the identification, analysis and description of the structure of morphemes and other units of meaning in a language like words, affixes, and parts of

speech and intonation/stress, implied context. As it is discussed by Lloret (1997), it is a process that involves Assimilation, Deletion, Epenthesis and Metathesis.

- ✓ **Assimilation**

Assimilation is a process where the two segments at a morph boundary influence each other, resulting in some feature change that makes them more similar (Lloret, 1997). As we can see below, both progressive as in [2] and regressive as in [1] and [3] types of consonant assimilation involving many of the consonants are taken place in the language. This can be partial assimilation in voicing [1] or manner [2] or it can be total assimilation as in [3].

[1] /gub-t-e/ →[gubde] /t/ →[d]/b-

Burn 'burnt'

[2] /gub-n-e/ →[gumne] /b/ →[m]/-n

Burn 'we burnt'

[3] /gal-n-e/ →[galle] /n/ →[l]/l-

Enter 'we entered'

- ✓ **Deletion**

Segment deletion, especially for vowels, takes place at morpheme boundary when two identical vowels come together as in [4] and [5]. As it is indicated by Grage (1976) and

Lloret (1988), such deletion process sometimes takes place as a mechanism of conforming to the phonotactic or syllable structure constraints of the language.

[4] /nama-i ch /ʧ/ a/→/nami ch /ʧ/ a/ /a/→-i

Man 'the man'

[5] /harree-oota/→/harroota/ /ee/→oo

Donkey 'donkeys'

- ✓ **Epenthesis**

Epenthesis is also another process that operates in the language. As it is shown in the example below, the vowel /i/ is epenthetically inserted between /g/ of the stem and /n/ of the suffix to prevent clustering of three consonants that is phonotactically impossible in the language (Lloret, 2005).

e.g. / 'arg-n-/ →['argine]

see-past 'we saw'

- ✓ **Metathesis**

A phonological process in which, for variety of reasons, letters, sounds and even syllables within a word are transposed (Wardhaugh, 1977).

In this chapter we tried to see the Afaan Oromo language characteristics to be noted in designing the TTS system for the language. The up-coming fourth chapter deals with the different tools that can be used to build the synthesizer and it also contains different works done by foreign and local researchers.

Chapter Four

The TTS Algorithm

This chapter is all about the basic TTS system algorithms used for the research work undertaken on Afaan Oromo language. In this study a TTS sytem is designed following the principle of concatenative synthesis.

According to Edgington and Lowry (*n.d.*), the basis of concatenative synthesis is to join short segments of speech, usually taken from a pre-recorded database, and then impose synthetic prosody (primarily pitch and duration) by appropriate signal processing. Both of these steps can introduce distortion to the synthetic speech: At the boundaries between speech segments by inappropriate selection or insufficient merging of segments, and by the prosodic modification process, due to insufficient robust speech modification model.

Therefore, to remedy the above mentioned distortions, Linear Predictive Coding which is a popular technique for speech compression and speech synthesis are used.

4.1 Overview of Linear Predictive Coding

It is a tool used mostly in audio signal processing and speech processing for representing the spectral envelope of a digital signal of speech in compressed form using the information of a linear predictive model. It is one of the most powerful speech analysis techniques, and

one of the most useful methods for encoding good quality speech at a low bit rate and provides extremely accurate estimates of speech parameters. Under normal circumstances, speech is sampled at 8000 samples/second with 8 bits used to represent each sample. This provides a rate of 64000 bits/second. Linear predictive coding reduces this to 2400 bits/second (Bradbury, 2000).

According to Honda (2003), human speech is produced in the vocal tract which can be approximated as a variable diameter tube. The linear predictive coding model is based on a mathematical approximation of the vocal tract represented by this tube of varying diameter. At a particular time, *t*, the speech sample *s(t)* is represented as a linear sum of the *p* previous samples. The most important aspect of LPC is the linear predictive filter which allows the value of the next sample to be determined by a linear combination of previous samples. It can also be stated as finding the coefficients a_k which result in the best prediction that minimizes mean-squared prediction error of the speech sample s[n] in terms of the past samples s[n−k], k={1,…,P}. The predicted sample s[n] is then given by:

$$s[n] = \sum_{k=1}^{p} (a_k s[n-k]) \qquad \textbf{4.1}$$

where P is the number of past samples of s[n] which we wish to examine.

There are different algorithms commonly used in LPC as discussed below.

4.2.2 Time Domain Pitch Synchronous Overlapp Add (TD-PSOLA)

Currently, TD-PSOLA is the most popular synthesis algorithm for text-to-speech systems. The algorithm produces very high quality synthetic speech, particularly when a pitch modification factor is small. However, as the pitch modification factor becomes larger, the quality degradation due to a slight pitch epoch detection error becomes severe. Although TD-PSOLA provides good quality speech synthesis, it has limitations which are related to its *non-parametric* structure; spectral mismatch at segmental boundaries and tonal quality when prosodic modifications are applied on the concatenated acoustic units.

According to Syrdal et al (1998), the TD-PSOLA algorithm was implemented in TCL scripting language. As a result of belonging to the group of interpreted languages, TCL is relatively slow. Even though the program was optimized, the synthesis of a sentence lasts several seconds.

The Multi-Band Resynthesis Overlap-Add (MBROLA) which is also one of the techniques used in a speech synthesis tries to overcome the TD-PSOLA concatenation problems mentioned above by re-synthesizing voiced parts with constant phase and constant pitch. But, for the reasons that the process is Artificial[6], it is not the main focus area for this specific

[6] **Artificial** →because re-synthesizing of the voice with consonant Phase and Pitch should have to be done manually.

work and detailed discussions will not be given; for that matter, it will always be the main source of MBROLA's problems, like fuzziness.

4.1.2 Residual Excited Linear Predictive Coding

Residual Excited Linear Predictive Coding is also one of the smoothening techniques used in speech technology. In this technique, the measure of similarity between signals which we call it Correlation will be calculated and each pitch period in the residual will be searched and isolated.

An LPC analyzer, a residual encoder, a residual decoder, a spectral flattener, and an LPC synthesizer are the basic components of RELP. As it is required by this method, pitch marks, Linear Predictive Coding (LPC) parameters and LPC residual values had to be extracted for each diphone in the diphone database (Wasala, et al. 2007). Then after, the correlation between two discrete time signals x[n] and y[n] is calculated as:

$$r_{xy}[l] = \sum_{n=-\infty}^{\infty} (x[n]y[n-l]) \quad \text{.....} \quad \textbf{4.2}$$

Where the sample index is denoted by n, and the time shift between the two signals is denoted by *l*.

Since speech signals are not stationary, we are typically interested in the similarities between signals only over short time duration. In this case, the cross-correlation is computed only over a window of time samples and for only a few time delays l= {0, 1…, P}. In the mean time, the redundant signals that may arise due to the similarity between neighboring signals during the wave generation can also be calculated as:

$$r_{ss}[l] = \left(1/N \sum_{n=0}^{N-1} (s[n]s[n-l]) \right) \quad \text{.....4.3}$$

Where s[n] stands for the speeches signal that exist, n= {-P, (-P) +1… N−1} are the known samples and the 1/N is a normalizing factor.

The difference between the predictor and the original signal is referred to as the error signal, also sometimes called the residual error, the LPC residual or residual excited linear prediction (RELP), or the prediction error and is given by:

$$\varepsilon[n] = s[n] - \bar{s}[n] = s[n] + \sum_{i=1}^{p} - a_t s[n-i] \quad \text{.....4.4}$$

Where $\bar{s}[n]$ stands for the predicted linear signal.

For the reasons that it needs sequences of residual signals for exciting the vocal tract model synthesized from speech signals, its compression rate is moderate. Moreover, the quality of synthesized speech is superior to other kinds of LPC and the system produced is robust

since there is no need to analyze whether the sound is voiced or unvoiced to analyze the pitch period.

Rowden (1992) stated that RELP technique allows the residual to be coded by conventional waveform coding techniques such as Pulse Code Modulation (PCM), Differential Pulse Code Modulation (DPCM), or Delta Modulation (DM). He also stated that, if the bandwidth of the residual is the same as for the original speech, the coding advantage comes from the reduced variance of the residual, needing fewer bits to achieve a given signal to noise ration. A very natural sounding speech can be produced, although this increase the total bit rate to 16 Kbit/s.

In general, the LPC analysis is used for two purposes. First, it allows separating the source (excitation) from the filter (formants) within the limits of linear model. Applying pitch and duration modification to the LPC residual allows introducing less distortion in modified speech. Second, the LPC coefficients are used for evaluating the spectral distance between two diphones during diphone candidate selection and between two triphones during triphone candidate selection at synthesis.

The above mentioned theoretical backgrounds, advantages and also the availability of the techniques in the publicly distributed Festival TTS system are the reasons why RELP is used for this specific thesis work. As required by this method, pitch marks, Linear Predictive

Coding (LPC) parameters and LPC residual values are extracted for each diphone and triphones in the database constructed.

The algorithm discussed here is also used to synthesize the Afaan Oromo speech units selected to build the TTS system for the language. The TTS system is developed using Festival architecture that uses Residual LPC technique to synthesize continuous words. The next chapter deals with the detailed descriptions of how the Afaan Oromo language TTS system is developed.

Chapter Five

Experimentations and Evaluations

In this research work a TTS system for Afaan Oromo language is designed and implemented. In this chapter specifically, the design and implementations of the built system is discussed in detail. In doing so, a synthesizer for both diphone and triphone prototype are designed and finally evaluations are done to test the performance as to the data set given.

5.1 Architecture of the Afaan Oromo Synthesizer

Basically there are three main methods that are used to build TTS synthesizer for Afaan Oromo: the Natural Language, the Digital Signal Processing Modules and the Databse construction. The Natural Language Processing Module is the first process where Text Analysis, Phonetic Analysis, Prosodic Analysis and Speech Analysis of the input texts are processed. Then, the Digital Signal Processing Module takes an action accordingly to render a voice after it received recorded utterances from the database; where this database is the set of Diphones and Triphones designed during the Database Constructions of the last component of the synthesizer.

The picture depicted in Figure 5.1 is all about the Afaan Oromo language TTS system built. As it is shown, before the input to be synthesized is given an utterance, it first passes the basic functions of the synthesizer.

The text analysis phase, within natural language processing module, converts non standard words to standard ones. The phonemic analysis phase which takes tokenized words is a text-to-phone or grapheme-to-phoneme converter, converting the written text into a sequence of phonemic symbols. The prosodic analysis module then takes the phoneme sequence, and assigns to each phoneme the required pitch and duration. Both the phonemic and prosodic analyses are typically language dependent. Then, finally, digital signal processing module accepts individual phonemes associated with their prosodic information and then produce synthesized speech, which is the outcome of the process.

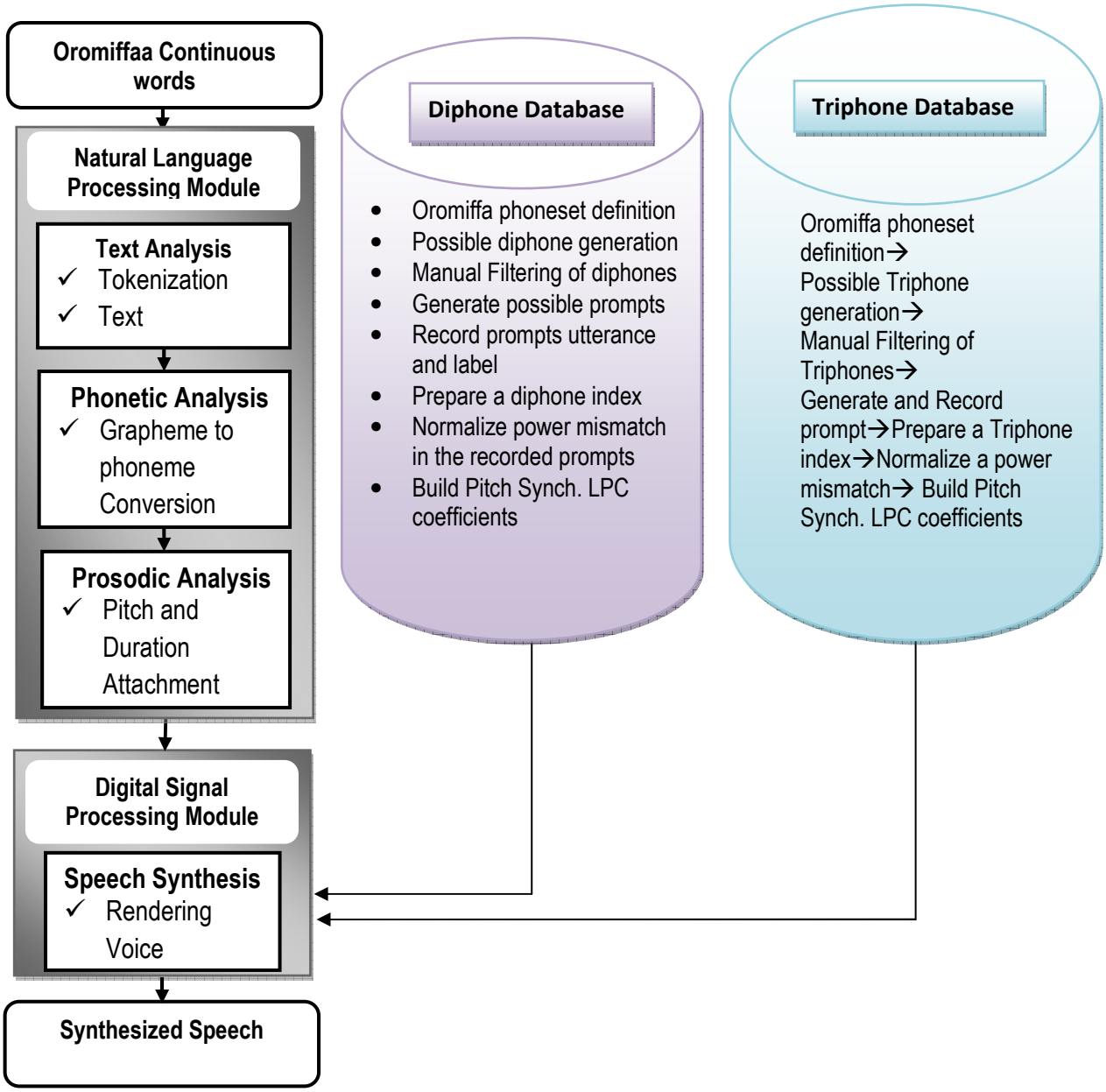

Figure 5.1 TTS system for Afaan Oromo

5.2 Text Analysis

One of the processes here in Afaan Oromo language TTS system development is the analysis of given text; where it totally depends on the total sum of tokenization and normalization. Tokenization is the process of breaking sequences of sentences into its constituent words. During tokenization, the white space delimiter and special characters (like ?,;…) are the main focus areas where, whenever there exists the mentioned delimiters between characters in a sentence, sequences of characters are broken to produce a meaningful word for a given specific language. For example, “Inni nama gaarii dha” (He is a Good Person) is tokenized as “inni”, “nama”,”gaarii”, and”dha”. The white space is used as one of the delimiters that is mentioned often and creates an utterance boundary; here then, the script was written so as to break a token from the source input sentence whenever a white space exists and goes on till it reaches the end of the sentence. Though the white space is the most commonly used delimiter between words and is extensively used for tokenization, it has got some limitation; a token type which allows the occurrence of white-space within the token will not recognize as a single token, but split up into two or more tokens. For example, consider a telephone number “0912 12 69 99”. This can identify as a single token of type ‘telephone number', but if tokenization is exclusively based on white-space, then we end up having 4 tokens. In addition, has it been other language, Japanese or Chinese, the usages of the white space delimiters to tokenize the input sentence would

totally be neglected for the reason that we don't get the delimiters in the languages (Black and Lenzo, 2000). Therefore, to remedy this limitation, every token is then has to go through a token identification process that identifies its token type/category.

In general, the text analysis subsystem extracts the linguistic and prosodic information from the input text given. The program iterates through the input text and extracts the gemination and other marks, and the sequences of syllables using the syllabification rule.

5.3 Phonetic Analysis

Phonetic analysis is a process where phone sets that are features best describing the Afaan Oromo language are described after the input text is normalized. The Oromiffaa phone sets were built using the lexical analysis which is supported by Festival system and the letter to sound modules. The letter-to-sound module is used here to convert an orthographic text into its corresponding phonetic representation. This module helps to match sounds of a phone whose utterances are not listed in the Lexicon. The lexicon in other case is a subsystem where the lexical analyses were performed. The language Afaan Oromo as one of the other languages with lots of loan words[7], to be pronounceable in the system, the letter-to-sound rule has been formulated specifically for the sample words used for this work.

[7] For example: like "koppii" (copy) and "talavidzinii" (television)

From the very beginning, the first option the festival system uses for a new instance is the Adenda Mechanism. This mechanism is effective for a very short instance introduced to the adenda because of linear searching mechanism that makes everything easy; this is then mentioned as a core reason to make it less efficient as compared to the compiled lexicon for searching is binary. Here, every word is mapped to its exact pronunciation in the lexicon and whenever any new instance doesn't appear to match to the pronunciations in the lexicon, it will automatically be forwarded to the letter-to-sound matches formed.

In addition, in Afaan Oromo language, geminated words and also words that have double vowels, though they seem to have the same property with words with a single consonant and vowel, they should be mapped to their unique property by hand than living them with the automatically generated features.

The architecture the researcher used here is the Grapheme-to-Phoneme (G2P) conversion architecture for the reasons that it is supported in Festival system and Oromiffaa being a phonetic language has an almost one-to-one mapping between letters and phonemes.

But the problem lies when there is a need to represent all words to its constituent pronunciation for it needs large corpus be classified according to the rules set.

5.4 Prosodic Analysis

One of the part of speech synthesis systems that depend on the nature of the language is the Prosody of a language. The prosody characteristic of a language mainly deals with the way words are spoken. With the goal of providing more prosodic and expressive control over unit selection TTS for dialog applications, while retaining naturalness, we have focused on speech acts, the communicative function of an utterance. The current working set of speech acts being used include: Imperative (directive, request, wait, repeat, and warning), Interrogative (question-wh, question-yes/no, and question-multiple choices), Assertive (informative-general, informative-detail), Affective (apology, exclamation-positive, exclamation-negative, greeting, good-bye, thanks) and others (confirmation, disconfirmation, back-channel, cue phrase). For this specific work, the first two are fully controlled where as the others are not controlled fully for the reason that it needs further investigation on the relationship between the data sets and the special characters.

Prosody indeed is the main component of any TTS system. Changing the placement of emphasis in a sentence can change the meaning of a word, and this emphasis might be revealed as a change in pitch, volume, voice quality, or timing. This is a place where the intonation and durations of every diphone or triphone units of any word will be tracked. In any language a word may differ contextually because of its dependent nature or because

there is a punctuation mark attached. For example, “Sirrii dhaa” is equivalent to saying “It is right” but whenever the punctuation mark ‘?’ is put with it like “Sirrii dhaa?” the previous normal sentence is changed to “Is it right?” which is a question type.

5.5 Database Construction

For this specific research a database with sets of Diphones and Triphones are prepared; where a diphone is created when the last half of the first monophone combines with the first half of the second monophone as it is mentioned in Section 2.4.3.

To start to develop the Oromiffaa database, first a phone-set that consisted every phones of the language are prepared. The proposed set of symbol scheme is found to be a versatile representation scheme for Oromiffaa phone-set. Along with the phone symbols, features such as vowel height, place of articulation and voicing are defined. Apart from the default set of features, new features that are useful in describing Oromiffaa phones are also defined. For example, whether a consonant is pre-nasalized or not, whether a phoneme is Ejectives or Implosive has been defined in there. These features are indeed proved extremely useful when implementing prosody. After this all done the finalized phone-set is placed in */home/sammy/festvox/oro/festvox/aau_oro_sam_phoneset.scm.* This phone-set is later used for both the diphone and triphone database construction whose detail is given below.

5.1.2 Diphone Database Construction

The Oromifaa language has 24 native basic consonants and five short and five long vowels as it is mentioned in Section 3.3.1. To construct the diphone database, the possible 34X34 phone-phone pairs, that is 1156 diphones were generated automatically using the command *Festvox/diphlist.scm Festvox/oro_schema.scm\ '(diphone_gen_schema "oro" "etc/orodiph.list")'* which is supported by the festival system; where *"orodiphlist.list"* stores list of the automatically generated Oromiffaa diphones. But from this all, since there existed diphones that don't totally exist in the language, the researcher selected eight hundred (800) combinations that represent the most frequent used words in the Oromo language randomly from daily used words that could be found in Newspapers, Books, and Streaming media after consulting linguists from AAU in the area. In Oromo language for example, we don't have the "hh" pair; therefore, this consonant-consonant combinations and others that don't represent words in the language have been eliminated from the constructed diphone database. In addition to this, since Oromiffaa language has a unique property than other local language in incorporating the quotation mark (') called "Hudha" in words, those generated diphones with consecutive two or more vowels have been edited manually with this special symbol so that words formed from this consecutive vowels give meaning. For example, the word *"gaa'ela"* whose equivalent English meaning is *"Mariage"* can be

mentioned. Appendix E shows some of the necessarily used diphones selected out of 1529 hypothetically generated diphones.

As it is shown in the Appendix E**,** every generated diphone has a unique ID that starts with *"oro_"* and owns a number according to their occurrence in the list. For example, *"oro_0086"* is to indicate that, the diphone 'o-s' is accessed with an ID labeled by *"oro_0086"*. This is just to keep the standardization and easily access the diphones during the process.

For the reasons that the researcher is familiar with the language, recording has been done with the voice of the researchers in a sound environment for an approximate time ranges of two hours. The researcher tried to record with Linux (specifically Red Hat 4.0) built-in voice recording tool. Unfortunately, the recorded sound was a low quality level with what has been then recorded with a freely available tool called Praat3; and then those recorded voices were kept in a *"/home/Sammy/festvox/oro/wav/"* directory that will be used for database building and LPC extraction. To produce a sound quality, a stereo mode has been then selected with 16 kHz frequency level at 16bit rate.

During recording, there existed instability of diphones speeches because of computers internal circuitry, microphones used and other unknown reasons. But a solution called a carrier phrase which is also supported by festival system has been used as mechanism to keep the consistency among the recorded diphones. What really the carrier phrase is, it is

just a group of diphones that encloses the diphones needed to be recorded; it was an unaccented nonsense word, pronounced with a steady intonation. By putting the diphone in the middle of other phones, we keep utterance-final lengthening or initial phone effects from making any diphone louder or quieter than the others. We used different carrier phrases for consonant-vowel, vowel-consonant, phone-silence, and silence-phone sequences.

After recording the diphones, we have also labeled to estimate the approximate positions of each phones and also segmented the two phones that make up each diphone. This was done automatically with a command *"bin/make_labs prompt-wav/*.wav"*. But it was not completely accurate at finding phone boundaries, and so automatic phone segmentation was hand-corrected and then stored with a file name that exactly tells it is a labeled file. For example, *"oro_0011.lab"* is a labeled file that is found at the 11th of the queued files stored in the database.

When the diphone is being labeled, LPC parameter and residual for diphone are extracted from each diphone file with carrier. This is because the carrier can be neglected from the database, and only the diphone is required for building the diphone database (Black and Lenzo, 2000). But before extracting the LPC parameter and residual, the pitch mark is marked; this pitch mark is important for building a pitch synchronized LPC or residual

excited LPC (RELP). This is done by using the pitch mark tools in Speech tools. The command used for extracting pitch mark for each diphone is: *"bin/make_pm lar/*.lar"*

5.5.2 Triphone Database construction

Although selecting diphones gives a fair range of phonetic variation, it doesn't give the full prosodic or even spectral range that is really desired. If we could select longer units, or more appropriate units for a database of speech, the resulting concatenated forms are not only better, they also require less signal processing to correct them to the targets (signal processing introduces distortion so minimizing its use is something that is inevitable). For this specific work, we tried to minimize the concatenation points that existed in diphone by moving to one higher phone level to provide a better speech quality; thus we have tried to build a database of triphone entries for Oromiffaa language to see the effects of the phone units on the language.

To construct the triphone database, there could have been used three kinds of techniques besides techniques suggested by Blomberg and Elenius (*n.d.*); concatenating two diphones, concatenating a diphone with a monophone and concatenating three monophones. But for this specific research, the researcher used the last technique which is concatenating three monophones; for monophones are already there starting from diphone database construction.

The combinations of the monophones of the language which is 34x34x34 were considered during triphone database construction. It would have been possible to use the diphones generated and combine with a monophone so as to reduce the size of the triphone database but it was beyond the researcher's potential to control the entries. Like that of diphone, here also, the triphones have been generated automatically with a command *Festvox/triphlist.scm Festvox/oro_schema.scm \ '(triphone_gen_schema "oro" "etc/orotriph.list")';* where *"orotriph.list"* stores the generated triphones.

But the constructions of the Triphone had not been as such mere combinations of phones. Rather it demanded a skillful knowledge to the scheme and thorough investigations of rebuilding of the codes in the festival system.

Here again, those triphones that don't represent the language, were selected and discarded from the database where as the triphones that do represent the language were selected and used as a potential resource for the database construction. For this specific research work, a total of *39304 triphones* were expected to be generated and used for the database construction. But since there are entries that don't represent the language[8], and using all the

[8] In Afaan Oromo language for example, three consonant phonemes are not allowed to go together. For example,"*lkj"* are not allowed to exist so they are removed from the database.

generated triphones as a set of the triphones increases the running time of the system and reduces the efficiency of the system, a total of 1982 triphone entries are used.

After the constructions of the database of the Diphone and Triphones the next step followed is Digital Signal Processing which is the last step in this Oromiffaa Speech synthesis.

5.6 Digital Signal Processing

In this module a transformation of the received symbolic information, from the NLP module, into speech is done. The main focus of the digital signal processing of the Afaan Oromo language TTS sytem is the concatenations of diphones and triphones to generate the wave form of a speech. The speech output is produced by coupling segments from the database to create the sequence of segments where these segments of speech are tied together to form a complete speech chain.

Concatenating the speech waveform results some glitches at the concatenation points in the synthesized utterances (Stylianou, 2001). Therefore, to ensure smooth concatenation of speech waveforms and to enable prosodic modifications on the speech units a speech model is generally used for speech representation and waveform generation.

Therefore, the attempts made to build the Oromiffaa TTS system using diphone and triphone speech units considers one of the frameworks of festival system: Residual Linear Predictive Coding mechanism, to smooth the glitches found during the utterances.

From the very beginning in this work, LPC parameters and LPC residual files for each file in the diphone and triphone databases are extracted and stored. Ideally, the LPC analysis should be done pitch-synchronously; thus, requiring that pitch marks are created before the LPC analysis takes place. This is done by first finding the pitchmark place of the diphone and triphone files, then passing the files together with their pitchmark through a LPC analysis to general pitch synchronize LPC parameters and LPC residual files for each diphone and triphone. This indeed shows that the diphone concatenation is not a mere collection of diphones that forms a word.

To concatenate the speech units that are used here, the amplitudes of the individual units forming a word are taken into considerations; their aplitudes needs to be zero or low at their juncture so that they can be aligned together. This means that as far as the diphones are concerned, the pitch periods at the end of the first diphone must line up with the pitch periods at the beginning of the second diphone; where as the triphone considers the alignement of the pitch periods at the end of the first triphone with the beginning of the third triphone and the alignement of the pitch periods at the end of the second triphone

with the beginning of the third triphone. Otherwise, the resulting single irregular pitch period at the juncture of each speech units is perceptible as well.

One of the techniques to reach such amplitude is a windowing function. For example, the diphone “fi” and ”ra” making the word “fira” (English equivalent is “relative”) have different wave signal at their juncture. We therefore, applied a windowing function to make the samples have low or zero amplitudes at their juncture.

For the reasons that the diphone and triphone databases are created individually, pitch marks of the speech units are passed independent of the other.

Having such information about the speech units, the DSP component then produces a speech that can be heard through the speakers of the computer.

5.7 Testing and Evaluations

Once the Oromiffaa Text To Speech system is developed, the system is then evaluated from two dimensions: from the researcher’s side and from the native to the Oromiffaa language’s side. For the reasons that the system incorporates both Diphone and Triphone databases, evaluations are done for each database. The general form of the evaluations is based on the Intelligibility and the Naturalness of the system.

The first experiment is done on word level from the researcher's point of view. The system is first given words[9] (Appendix H lists 60 standard words used) selected with the help of a domain expert from Addis Ababa University Institutes of Language Studies and a book on Oromiffaa grammar written by Takilee (2010). Values are then given based on the rate of recognition of words by the system; this rate is directly related to the performance of the system. Table 5.1 shows the performance measurement of the system.

	Performance Measurement Values Given					
	Pronounced Correctly		**Pronounced Partially**		**Incorrectly Pronounced**	
	In Number	***In Percent (%)***	***In Number***	***In Percent (%)***	***In Number***	***In Percent (%)***
Words (Diphones)	60	75	13	16.25	7	8.75
Words (Triphones)	43	54	25	31	12	15

Table 5.1 The Performance measurement of Oromiffaa TTS system

The above result explicitly shows that, the performance measurement of the system for diphone speech units is 75% where as for the Triphone its performance is 54%. The result found for triphone speech unit is encouraging for further investigations on longer units. The result indeed is predicting the increasing of the result to much higher level if further modifications are taken place.

[9] These words are words used for testing both the Diphones and Triphones so as to compare the performance of the system on each speech units.

The more the testing data, the higher the result would have been for both the diphones and the triphones. The result of the triphone indicates that, the system couldn't locate those tried words which might have been removed because of the time complexity mentioned above.

As it is observed from the analysis, words that are not correctly pronounced are those that are not found in the compiled lexicon. What the festival system does first is, it matches the input words pronunciation with those words found in the compiled lexicon, then the letter to sound rule is evoked if no match is found to pronounce the word.

Basically, the letter to sound rule has entries that match each letters in the language alphabet to some possible sound forms that the letter can assume. This demands consideration of all the possible sounds in a large set of words (thousands of words) in the language. However, in this work, since the letter to sound rule is constructed based on the chosen 200 words, it results in improper pronunciation of the words not in the compiled lexicon. This outlines the main adverse effect in degraded performance of the system.

The other form of system evaluation focuses on the Intelligence and Naturalness character of the system. Six native speakers of the language are invited to rate the systems performance based on the selected six sentences (see the detail in APPENDIX G) with a

MOS scale levels after they are given a questionnaire (details are in APPENDIX A). For the reasons that it is simple and commonly used technique for overall TTS system evaluations, here also MOS evaluation technique is used so that evaluators assess the system and use MOS scale level depicted in Table 5.2.

Values	MOS
5	Excellent
4	Very Good
3	Good
2	Fair
1	Bad

Table 5.2 MOS scale level

For the utterances of the system for the input sentence, the evaluators rate based on the MOS scale level given in Table 5.2. The result of the Intelligence and Naturalness evaluations of the system is shown in Table 5.3 and 5.4 consecutively.

Test data (Sentences)	Ranks that ranges from 1 to 5 given by evaluators to test the Intelligibility												Average	
	P 1		P 2		P 3		P 4		P 5		P 6			
	D	*T*	*D*	*T*	*D*	*T*	*D*	*T*	*D*	*T*	*D*	*T*	*D*	*T*
1	3	2	3	2	3	2	3	2	3	1	3	2	**3**	**1.8**
2	3	3	2	2	3	3	2	1	3	3	3	2	**2.7**	**2.3**
3	3	2	2	3	3	2	3	2	3	2	4	2	**3**	**2.2**
4	4	3	3	3	4	3	3	2	3	3	4	3	**3.5**	**2.7**
5	2	2	3	2	3	1	4	3	3	2	3	2	**3**	**2.2**
6	3	3	3	1	4	2	3	2	1	2	3	2	**3**	**2**
Mean Average													**3.05**	**2.2**

Table 5.3 Results of measurements of the Intelligibility of the system
(D=Diphone and T=Triphone)

Test data (Sentences)	Ranks that ranges from 1 to 5 given by evaluators to test the Naturalness												Average	
	P 1		P 2		P 3		P 4		P 5		P 6			
	D	*T*	*D*	*T*	*D*	*T*	*D*	*T*	*D*	*T*	*D*	*T*	*D*	*T*
1	3	2	3	2	3	2	3	2	3	2	3	2	**3**	**2**
2	3	2	3	2	3	1	3	2	3	2	2	2	**2.8**	**1.8**
3	3	1	3	3	2	1	3	1	2	2	3	2	**2.7**	**1.7**
4	3	3	2	2	3	2	3	2	2	2	3	2	**2.6**	**2.2**
5	3	2	2	3	2	2	2	3	3	1	2	2	**2.3**	**2.2**
6	3	2	3	3	3	2	2	2	2	2	2	2	**2.5**	**2.2**
Mean Average													**2.65**	**2.02**

Table 5.4 Results of measurements of the Naturalness of the system
(D=Diphone and T=Triphone)

Based on the randomly selected six sentences and six evaluators, the intelligibility of the system as it can be seen from the Table 5.3 above is 3.03 and 2.2 for diphones and triphones respectively. As to the MOS scale level given, the intelligibility of the system for the diphone speech unit is rated under "good" level; while the intelligibility of the system is rated under "fair".

However, the Naturalness of the system based on the result found in Table 5.4 is 2.65 and 2.02 for diphones and triphones respectively. As a matter of fact, the system again is rated under "good" for naturalness based on the MOS scale given to the diphones and rated under a "fair" for Triphones.

The possible reasons behind this result (for triphones) could be the removal of the triphone entries that are found to increase the running time of the system as it is mentioned in Section 5.6.2

Triphone	
Intelligence	***Naturalness***
1.9	**2**
2.3	**2**
2.3	**1.7**
2.7	**2.2**
2.2	**2.2**
2	**2.2**
2.23	**2.05**

Triphone	
Intelligence	***Naturalness***
1.9	**2**
2.4	**2.1**
2.3	**1.8**
2.7	**2.2**
2.2	**2.2**
2.1	**2.2**
2.27	**2.08**

Table 5.5 MOS results for the modified triphone database

As we can see in Table 5.5, as we try to modify the triphone database by adding some of those entries removed earlier, the intelligibility of the system increases accordingly. This shows that there appears to be a strong relationship between the consecutive three phones

of the Oromiffaa language and indeed the result found is also encouraging to come up with a much promising result as we go up modifying the database.

The above findings illustrate to give a more focus on the entries of the database of the diphone and triphones to come up with a higher result than found here.

Chapter Six

Conclusions and Recommendations

6.1 Conclusions

In this work, the researcher has shown that, building a TTS for Oromifaa language using diphone and triphone speech units is possible. The longer the selected units are, the fewer problematic concatenation points occur in the synthetic speech. Though memory requirement used to be the only question to answer for longer units in TTS, right this time this question is not the main concern behind; that is indeed why concatenation using longer units specifically triphones, are tried here.

The Residual Excited Linear Predictive Coding (RELPC) which is one of the building blocks of the festival system and the three major areas of any speech synthesis: database construction, natural language processing and digital signal processing are used to build the Oromiffaa TTS synthesizer; where triphone speech units are given more focus for this specific research work.

Finally, evaluations are done individually for each speech units so that comparison between the speech units would be done. Then after, a result of 75% and 54% is found to be the performance of the system based on the diphone and triphone speech units respectively. A MOS result for Intelligibility of the system is found as 3.03 and 2.2 for the diphone and

triphone speech units respectively; whereas, the Naturalness of the system is found to be 2.65 and 2.02 based on the diphone and triphones speech units.

As it can be shown in the findings, the fourth sentence (see the detail in Appendix G) is understandable by the listener easily than others; the values given to the utterances of the triphones and diphones tell explicitly. This indeed has a direct relationship with the number of concatenation points of the speech units; as the number of the concatenation points become few the more the system preserves the natural sounds of the speaker.

This first attempt to develop a concatenative triphones TTS system for Oromiffaa language seems encouraging so that researchers can follow the trend and enact to extend its performance and try to investigate the relationships between higher level speech units even for other local languages.

This research work, though it can be taken as a strong asset for further research works that focuses more on the human computer interaction, it has also its own weakness that needs closer attentions. We tried to record the sound in a quite room but it still experienced a noise that disturbs the overall naturalness of the system; for the reason that the outcomes of the system is highly dependent on the input given. Had there been a sound quite laboratory, the naturalness of the system would have been higher.

Generally, the finding indicates that, diphones are the appropriate speech units for Afaan Oromo language with an accuracy of 75%. But there is also other new promising finding on triphone speech units with an accuracy of 54%; and a MOS scale levels reaching to good. From this finding, researchers on other local languages can be benefited more as they try to conduct a research on higher speech units.

6.2 Recommendations

- In this specific work NSW's are not included. To design a general TTS system for Afaan Oromo there is a need to include them so as to extend the range of the system.
- The focus of the next research work must also be on other forms of generating the triphones other than done here (such as, concatenating a diphone with a monophone can be one example); to further investigate the advantages of triphones and also minimize the database size.
- The next research work can extend its range to clusters of three consonant phones that are not allowed in the Afaan Oromo language, by inserting vowels epenthetically.
- For the reasons that a large corpus is not available for the Afaan Oromo language, it was difficult to select entries for training and testing the system. But it is promising that increasing the number of sentences and words to be tested gives more accurate values

than found here. Therefore, the incorporation of a large standardized corpus should be the focus area of the next research work.

- In this work, Homographs are not included for the reasons that it needs Symantec analysis than Orthographic analysis. Hence, there is a need to incorporate words similar in structure but different in context.
- Words with single quotations which we can find in the Oromiffaa language are not included in this specific work because it needs further investigations of the festival system to incorporate the rules of the Afaan Oromo language into it.
- For the reason that a diverse regional dialects appear within the Oromo people, further investigations has to be done on intonation differences so as to incorporate those different dialects to TTS.
- Finally, in this specific work, a trial has been made with a diphone and triphone units of speech. Further emphasis should be given on longer speech units to investigate the gap between each speech units.

Reference

Assaf, M. (2005). "A Prototype of an Arabic Diphone Speech Synthesizer in Festival". MSc. Thesis, Uppsala Universitet, Stockholm, Sweden.

Black, A.W. and Lenzo, K.A. (2000). "Building Synthetic Voices - for FestVox 2.0 Edition". Retrieved from: http://www.festvox.org/bsv/ [Accessed on: August, 2010]

Blomberg, M. and Elenius, K. (*n.d.*). "Creation of Unseen Triphones from Diphones and Monophones using a Speech Production Approach". Dept. of Speech, Music and Hearing, KTH, Stockholm, Sweden.

Bradbury, J. (2000). "Linear Predictive Coding". Retrieved from: http://www.my.fit.edu/~vkepuska/ece5525/lpc_paper.pdf [Accessed on: August, 2010]

Donovan, R. E. (1996). "Trainable Speech Synthesis". PhD Dissertation, University of Cambridge, England.

Dutoit, T. (1997). "An Introduction to Text-To-Speech Synthesis". Dordrecht: Kluwer Academic Publishers.

Gragg, G. (1976). "Oromo of Wellegga". The Non-Semitic Languages of Ethiopia ed. by Lionel Bender, 166-195. East Lansing: Michigan State University Press.

Hassim, S. (2000). "A Corpus Based Concatenative Speech Synthesis System for Turkish". BSc. Dept. of Computer Engineering and Information Science, Bilkent University, Turkey.

Henock, L. (2003). "Concatenative Text To Speech (TTS) system for Amharic language". MSc. Thesis, Addis Ababa University, Ethiopia.

Honda, M. (2003). "Human speech Production Mechanisms". NTT Technical Review. Japan Science and Technology, 1(2): 24-29.

Huang, X., Acero, A., Adcock, J., Hon, H.W., Goldsmith, J., Liu, J., and Plumpe, M. (1996). "Whistler: A Trainable Text-To-Speech System", 4: 2387-2390.

Javidan, R., and **Rasekh, I.** (2010). "Concatenative Synthesis of Persian Language Based on Word, Diphone and Triphone Databases". Journal of Modern Applied science: Universtity of Islamic Azad-Beyza Branch, Fars Iran, 4(10).

Jurafsky, D. and Martin, J. (2000). "Speech and language processing". University of Colorado. Boulder: Prentice Hall, Inc.

Klatt, D. (1987). "Review of text-to-speech conversion for English". Journal of the Acoustical Society of America, 82(3): 737-793.

Lemmetty, S. (1999). "Review of Speech Synthesis Technology". MSc. Thesis, Helsinki University of Technology, Finland.

Lloret, M. (1988): "Gemination and vowel length in Oromo morphophonology". Ph.D. dissertation, Indiana University, Bloomington, India.

Lloret, M. (1997). "Oromo Phonology". Phonologies of Asia and Africa ed. by Alan S. Kaye, Winona Lake, Ind. Eisenbrauns, 1: 493-519.

Lloret, M. (2005). "Revising the phonological motivation for splitting the morpholpgy". Departament de Filologia Catalana Universitat de Barcelona, Spain.

Mc Tear, F. (2002). "Spoken Dialog Technology: Enabling the Conversational user interface". 34(1): 91-169.

Morka, M. (2001). "Text To Speech of Afaan Oromo". MSc. Thesis, Addis Ababa University, Ethiopia.

Ng, K. (1998)." Survey of data-driven approaches to speech synthesis". Retrieved from: http://www. citeseer.nj.nec.com/ng98survey.html [Accessed on: September, 2010]

Rabiner, L. (1993). "Fundamentals of Speech Recognition". Prentice Hall, Inc., USA.

Raj, A. A., Sarkar, T., Pammi, S. C., Yuvaraj, S., Bansal, M., Prahallad, K., and **Black, A. W.** (2007). "Text Processing for Text-to-Speech Systems in Indian Languages". Poc. 6th ISCA Speech Synthesis Workshop (SSW6), Bonn, Germany.

Rodman, R. D. (1999). "Computer Speech Technology". Artech House, Inc., London.

Rowden, C. (1992). "Speech Processing". UK: McGraw-Hill, Inc.

Schröder, M. (2005) "Emotional Speech Synthesis review". DFKI, Saarbrücken, Institute of Phonetics, University of the Saarland, Germany.

Sebsibe H., Kishore, S., Black, A. W., Kumar, R., and **Sangal, R.** (2004). "Unit Selection Voice for Amharic Using Festvox", ISCA Speech Synthesis Workshop, Pittsburgh: 103-107.

Stylianou, Y. (2001) "Applying the Harmonic Pulse Noise Model in Concatenative Speech Synthesis", IEEE Trans. on Speech and Audio Processing, 9(1): 21-29.

Syrdal, A., Stylianous, Y., Garrison, L., Conkie, A., and **Schroeter, J.** (1998) "TD-PSOLA versus Harmonic Puls Noise Model in Diphone Based Speech Synthesis", IEEE International Conference, 1: 273-276.

Retrieved from: http://www.research.att.com/projects/tts/papers/1998. [Accessed on: June 2010]

Swee, T. T. (2004). "The Design and Verification of Malay Text to Speech". M. Eng. Thesis, University of Technology Malaysia, Skudai, Malaysia. Retrieved from: http://sps.utm.my/sps/Academic%20Resources/abstract-of-thesis/2004/tan-tian-swee. [Accessed on: August, 2010]

Takilee, Q. (2010). "Seerluga Afaan Oromoo: Oromo Grammar". Addis Ababa, Ethiopia.

Tesfaye, Y. (2004). "Diphone Based Text-To-Speech Synthesis System for Tigrigna Language" Master's Thesis, Addis Ababa University, Ethiopia.

Tilahun, G. (1992). "Afan Oromo". Journal of Oromo Studies, Addis Ababa University, Ethiopia.

Vosnidis, C. (2001). "Speech Synthesis by Word Concatenation". B.Sc. Thesis, University of Crete, Greece.

Wasala, A., Weerasinghe, R., and **Gamage, K.** (2007). "A Sinhala Text-to-Speech System", Language Technology Research Laboratory, University Of Colombo School of Computing, Sri Lanka.

Wouters, J. (1996). “Analysis and synthesis of Degree of Articulation”. MSc. Electrical Engineering, Katholieke Universiteit Leuven, Belgium.

APPENDICES

A: Questionnaire

The aim of this questionnaire is to test the over-all performance of the Afaan Oromo language Text To Speech system. Below you can get two kinds of questions to be answered. Please feel free and try to complete all. I would also welcome the enthusiasm you have for the work undertaken.

Part 1

1. Select your current education level:

   1-8

   9-12

   >12

2. What is your nationality? _____

3. Are you familiar with Oromiffaa?

   Yes

   No

4. How do you rate your knowledge of the language (Oromiffaa):

Bad

Good

Very good

Part 2

Listen to the sound from the computer and rate based on the below given scale on the table.

5	**Excellent**
4	**Very-good**
3	**Good**
2	**Fair**
1	**Bad**

Sentence1._____

Sentence2._____

Sentence3._____

Sentence4._____

Sentence5._____

Sentence6._____

B: A Diphone Schema for Afaan Oromo language

```
;;; A diphone list for OROMIFFAA Language


(set! vowels '(a e i o u aa ee ii oo uu))
(set! consonants '(b c d f g h j k l m n q r s t w x y z ch dh ny ph sh))

 (set! silence 'pau)
;; for consonant clusters
(set! stops '(p b t d k g q ax))
(set! nasals '(n m nx))
(set! liquids '(l r))
(set! clusters1
      (append
       (apply
     append
    (mapcar (lambda (b) (mapcar (lambda (a) (list a b)) stops)) liquids))
       (mapcar (lambda (b) (list 's b)) '(l w n m p t k f))
       (mapcar (lambda (b) (list 'f b)) '(l r ))))
(set! clusters2
      (mapcar (lambda (b) (list b 'y))  (append stops nasals '(s f v))))

(set! cvc-carrier '((pau t aa ) (aa pau)))
(set! ooer-carrier '((pau t aa ) (aa pau)))
(set! vc-carrier '((pau t aa t) (aa pau)))
(set! cv-carrier '((pau t aa ) (t aa pau)))
(set! cc-carrier '((pau t aa ) (aa t aa pau)))
(set! vv-carrier '((pau t aa t) (t aa pau)))
(set! silv-carrier '(() (t aa pau)))
(set! silc-carrier '(() (aa t aa pau)))
(set! vsil-carrier '((pau t aa t ) ()))
(set! csil-carrier '((pau t aa t aa) ()))


(define (list-cvcs)
  (apply
   append
   (mapcar
    (lambda (v)
      (mapcar
       (lambda (c)
       (list
        (list (string-append c "-" v) (string-append v "-" c))
        (append (car cvc-carrier) (list c v c) (car (cdr cvc-carrier)))))
       consonants ))
    vowels)))
```

```
(define (list-vcs)
  (apply
   append
   (mapcar
    (lambda (v)
      (mapcar
       (lambda (c)
       (list
        (list (string-append v "-" c))
        (append (car vc-carrier) (list v c) (car (cdr vc-carrier)))))
     consonants))
    vowels)))

(define (list-cvs)
  (apply
   append
   (mapcar
    (lambda (c)
      (mapcar
       (lambda (v)
       (list
        (list (string-append c "-" v))
        (append (car cv-carrier) (list c v) (car (cdr cv-carrier)))))
       vowels))
     consonants)))

(define (list-ccs)
  (apply
   append
   (mapcar
    (lambda (c1)
      (mapcar
       (lambda (c2)
       (list
     (list (string-append c1 "-" c2))
     (append (car cc-carrier) (list c1 '- c2) (car (cdr cc-carrier)))))
        consonants ))
  consonants )))


(define (list-vvs)
  (apply
   append
   (mapcar
    (lambda (v1)
      (mapcar
       (lambda (v2)
       (list
        (list (string-append v1 "-" v2))
        (append (car vv-carrier) (list v1 v2) (car (cdr vv-carrier)))))
       vowels))
```

```
    vowels)))


(define (list-silv)
  (mapcar
   (lambda (v)
     (list
      (list (string-append silence "-" v))
      (append (car silv-carrier) (list silence v) (car (cdr silv-
carrier)))))
   vowels))

(define (list-silc)
  (mapcar
   (lambda (c)
     (list
      (list (string-append silence "-" c))
      (append (car silc-carrier) (list silence c) (car (cdr silc-
carrier)))))
   consonants))

(define (list-vsil)
  (mapcar
   (lambda (v)
     (list
      (list (string-append v "-" silence))
      (append (car vsil-carrier) (list v silence) (car (cdr vsil-
carrier)))))
   vowels))

(define (list-csil)
  (mapcar
   (lambda (c)
     (list
      (list (string-append c "-" silence))
      (append (car csil-carrier) (list c silence) (car (cdr csil-
carrier)))))
   consonants))


(define (diphone-gen-list)
  "(diphone-gen-list)
Returns a list of nonsense words as phone strings."
  (append
   (list-cvcs)  ;; consonant-vowel and vowel-consonant
   (list-vcs)  ;; one which don't go in cvc
   (list-cvs)  ;;
  (list-vvs)  ;; vowel-vowel
   (list-ccs)  ;; consonant-consonant
   (list-silv)
   (list-silc)
   (list-csil)
```

```
   (list-vsil)

   (list
    '(("pau-pau") (pau t aa t aa pau pau)))

   ))


(define (Diphone_Prompt_Setup)
 "(Diphone_Prompt_Setup)
Called before synthesizing the prompt waveforms.  Uses the KAL speakers
(the most standard US voice)."
 (voice_kal_diphone)  ;; US male voice
 (set! FP_F0 90)
 (set! diph_do_db_boundaries nil)    ;; lower F0 than ka;
 )
(set! nhg2radio_map
    '((a aa)
      (u uh)
      (o ao)
      (e ah)
      (i ae)
      (ee ah)
      (ie ay)
       (h ah ch)
       (ii ae)
      (oo ao ao)
       (uu uh uh)
       (ax t s)
      (tx t t)
      (sx s sh)
      (zx z sh)
       (ph p)
      (kx k hh)
      (hx sh)
       (x  t)
      (xx t sh)
      (c ch)
      (j jh)
      (cx ch hh)
      (ny n)
       (q k hh)
      (sil pau)))

(define (Diphone_Prompt_Word utt)
  "(Diphone_Prompt_Word utt)
Specify specific modifications of the utterance before synthesis
specific to this particulat phone set."
  ;; No syllabics in ked so flip them to non-syllabic form
  (mapcar
   (lambda (s)
     (let ((n (item.name s))
```

```
        (newn (cdr (assoc_string (item.name s) nhg2radio_map))))
       (cond
      ((cdr newn)  ;; it's a dual one

       (let ((newi (item.insert s (list (car (cdr newn))) 'after)))
         (item.set_feat newi "end" (item.feat s "end"))
         (item.set_feat s "end"
                   (/ (+ (item.feat s "segment_start")
                         (item.feat s "end"))
                      2))
         (item.set_name s (car newn))))
      (newn
       (item.set_name s (car newn)))
      (t
       ;; as is
       ))))
   (utt.relation.items utt 'Segment))
  utt)

(provide 'oro_schema)
```

C: The Afaan Oromo language Triphone Generator

```
(set! vowels '(a e i o u aa ee ii oo uu))

(set! consonants '(b c d f g h j k l m n q r s t w x y ch dh sh ts))

(set! silence 'pau)
(set! stops '(p b t d k g q ax))
(set! nasals '(n m nx))
(set! liquids '(l r))
(set! clusters1
      (append
       (apply
     append
    (mapcar (lambda (b) (mapcar (lambda (a) (list a b)) stops)) liquids))
       (mapcar (lambda (b) (list 's b)) '(l w n m p t k f))
       (mapcar (lambda (b) (list 'f b)) '(l r ))))
(set! clusters2
      (mapcar (lambda (b) (list b 'y))  (append stops nasals '(s f v))))

(set! ooer-carrier '((pau t aa ) (aa pau)))
(set! vcc-carrier '((t aa pau) (aa pau) (aa t aa pau)))
(set! vcv-carrier '((pau t aa t) (aa pau) (t aa pau)))
(set! vvc-carrier '((pau t aa t) (t aa pau) (aa pau)))
(set! vvv-carrier '((pau t aa t) (t aa pau) (t aa pau)))
(set! ccc-carrier '((pau t aa ) (aa t aa pau) (aa t aa pau)))
(set! ccv-carrier '((pau t aa ) (aa t aa pau) (t aa pau)))
(set! cvc-carrier '((pau t aa ) (t aa pau) (aa pau)))
(set! cvv-carrier '((pau t aa ) (t aa pau) (t aa pau)))
(set! silv-carrier '(() (t aa pau)))
(set! silc-carrier '(() (aa t aa pau)))
(set! vsil-carrier '((pau t aa t ) ()))
(set! csil-carrier '((pau t aa t aa) ()))

//……
(define (list-vccs)
  (apply
   append
   (mapcar
    (lambda (v)
      (mapcar
       (lambda (c)
       (list
        (list (string-append v "-" c) (string-append c "-" c))
        (append (car vcc-carrier) (list v c c) (car (cdr vcc-carrier)))))
       consonants ))
    vowels)))
```

```
(define (list-vcvs)
  (apply
   append
   (mapcar
    (lambda (v)
      (mapcar
       (lambda (c)
       (list
        (list (string-append v "-" c) (string-append c "-" v))
        (append (car vcv-carrier) (list v c v) (car (cdr vcv-carrier)))))
       consonants ))
    vowels)))

(define (list-vvcs)
  (apply
   append
   (mapcar
    (lambda (v)
      (mapcar
       (lambda (c)
       (list
        (list (string-append v "-" v) (string-append v "-" c))
        (append (car vvc-carrier) (list v v c) (car (cdr vvc-carrier)))))
       consonants ))
    vowels)))

(define (list-vvvs)
  (apply
   append
   (mapcar
    (lambda (v)
      (mapcar
       (lambda (v)
       (list
        (list (string-append v "-" v) (string-append v "-" v))
        (append (car vvv-carrier) (list v v v) (car (cdr vvv-carrier)))))
       consonants ))
    vowels)))

(define (list-cccs)
  (apply
   append
   (mapcar
    (lambda (c)
      (mapcar
       (lambda (c)
       (list
        (list (string-append c "-" c) (string-append c "-" c))
        (append (car ccc-carrier) (list c c c) (car (cdr ccc-carrier)))))
       consonants ))
    vowels)))
```

```
(define (list-ccvs)
  (apply
   append
   (mapcar
    (lambda (c)
      (mapcar
       (lambda (v)
       (list
        (list (string-append c "-" c) (string-append c "-" v))
        (append (car ccv-carrier) (list c c v ) (car (cdr ccv-carrier)))))
       consonants ))
    vowels)))

(define (list-cvcs)
  (apply
   append
   (mapcar
    (lambda (v)
      (mapcar
       (lambda (c)
       (list
        (list (string-append c "-" v) (string-append v "-" c))
        (append (car cvc-carrier) (list c v c) (car (cdr cvc-carrier)))))
       consonants ))
    vowels)))

(define (list-cvvs)
  (apply
   append
   (mapcar
    (lambda (v)
      (mapcar
       (lambda (c)
       (list
        (list (string-append c "-" v) (string-append v "-" v))
        (append (car cvv-carrier) (list c v v) (car (cdr cvv-carrier)))))
       consonants ))
    vowels)))

(define (list-silv)
  (mapcar
   (lambda (v)
     (list
      (list (string-append silence "-" v))
      (append (car silv-carrier) (list silence v) (car (cdr silv-
carrier)))))
   vowels))

(define (list-silc)
  (mapcar
   (lambda (c)
     (list
```

```
      (list (string-append silence "-" c))
      (append (car silc-carrier) (list silence c) (car (cdr silc-
carrier)))))
   consonants))

(define (list-vsil)
  (mapcar
   (lambda (v)
     (list
      (list (string-append v "-" silence))
      (append (car vsil-carrier) (list v silence) (car (cdr vsil-
carrier)))))
   vowels))

(define (list-csil)
  (mapcar
   (lambda (c)
     (list
      (list (string-append c "-" silence))
      (append (car csil-carrier) (list c silence) (car (cdr csil-
carrier)))))
   consonants))
```

D: The Afaan Oromo language Phoneset

```
 (defPhoneSet

aau_wo
;;;  Phone Features
(;; vowel or consonant
 (vc + -)
 ;; vowel length: short long dipthong schwa
 (vlng s l d a 0)
 ;; vowel height: high mid low
 (vheight 1 2 3 0)
 ;; vowel frontness: front mid back
 (vfront 1 2 3 0)
 ;; lip rounding
 (vrnd + - 0)
 ;; consonant type: stop fricative affricate nasal lateral approximant
 (ctype s f a n l r 0)
 ;; place of articulation: labial alveolar palatal labio-dental
 ;; dental velar glottal
 (cplace l a p b d v g 0)
 ;; consonant voicing
 (cvox + - 0)
 )
(
 (SIL -  0  0  0  0  0  0  -)  ;; slience ...
 (a  +   s  3  3  -  0  0  0)
 (e  +   s  2  1  -  0  0  0)
 (i  +   s  1  1  -  0  0  0)
 (o  +   s  2  3  +  0  0  0)
 (u  +   s  1  3  +  0  0  0)
 (aa  +  l  3  3  -  0  0  0)
 (ii  +  l  1  1  -  0  0  0)
 (ee  +  l  2  1  -  0  0  0)
 (oo  +  l  2  3  +  0  0  0)
 (uu  +  l  1  3  +  0  0  0)
 (ai  +  d  3  2  -  0  0  0)
 (ui  +  d  2  1  -  0  0  0)
 (oi  +  d  3  3  +  0  0  0)
 (au  +  d  3  2  +  0  0  0)
```

```
 (b - 0 0 0 0 s l +)
 (c - 0 0 0 0 a p -)
 (d - 0 0 0 0 s a +)
 (g - 0 0 0 0 s v +)
 (h - 0 0 0 0 f g -)
 (j - 0 0 0 0 a p +)
 (k - 0 0 0 0 s v -)
 (l - 0 0 0 0 l a -)
 (m - 0 0 0 0 n l -)
 (n - 0 0 0 0 n a -)
 (p - 0 0 0 0 s l -)
 (r - 0 0 0 0 r a -)
 (s - 0 0 0 0 f a -)
 (t - 0 0 0 0 s a -)
 (w - 0 0 0 0 r l -)
 (x - 0 0 0 0 a p -)
 (y - 0 0 0 0 r p -)
 (z - 0 0 0 0 f a +)

  )
)

(PhoneSet.silences '(SIL))

(define (aau_oro_sam::select_phoneset)
  "(aau_oro_sam::select_phoneset)
Set up phoneset for aau_oro."
  (Parameter.set 'PhoneSet 'aau_oro)
  (PhoneSet.select 'aau_oro)
)

(define (aau_oro_sam::reset_phoneset)
  "(aau_oro_sam::reset_phoneset)
Reset phoneset for aau_oro."
)

(provide 'aau_oro_sam_phoneset)
```

E: Sample Afaan Oromo language Diphone list

```
( oro_0241 "pau t aa t a b aa pau" ("a-b") )
( oro_0242 "pau t aa t a c aa pau" ("a-c") )
( oro_0243 "pau t aa t a d aa pau" ("a-d") )
( oro_0244 "pau t aa t a f aa pau" ("a-f") )
( oro_0249 "pau t aa t a l aa pau" ("a-l") )
( oro_0250 "pau t aa t a m aa pau" ("a-m") )
( oro_0251 "pau t aa t a n aa pau" ("a-n") )
( oro_0252 "pau t aa t a q aa pau" ("a-q") )
( oro_0276 "pau t aa t e q aa pau" ("e-q") )
( oro_0277 "pau t aa t e r aa pau" ("e-r") )
( oro_0278 "pau t aa t e s aa pau" ("e-s") )
( oro_0279 "pau t aa t e t aa pau" ("e-t") )
( oro_0280 "pau t aa t e w aa pau" ("e-w") )
( oro_0281 "pau t aa t e x aa pau" ("e-x") )
( oro_0282 "pau t aa t e y aa pau" ("e-y") )
( oro_0298 "pau t aa t i m aa pau" ("i-m") )
( oro_0299 "pau t aa t i n aa pau" ("i-n") )
( oro_0300 "pau t aa t i q aa pau" ("i-q") )
( oro_0301 "pau t aa t i r aa pau" ("i-r") )
( oro_0302 "pau t aa t i s aa pau" ("i-s") )
( oro_0303 "pau t aa t i t aa pau" ("i-t") )
( oro_0304 "pau t aa t i w aa pau" ("i-w") )
( oro_0305 "pau t aa t i x aa pau" ("i-x") )
( oro_0306 "pau t aa t i y aa pau" ("i-y") )
( oro_0307 "pau t aa t i z aa pau" ("i-z") )
( oro_0308 "pau t aa t i ch aa pau" ("i-ch") )
( oro_0309 "pau t aa t i dh aa pau" ("i-dh") )
( oro_0310 "pau t aa t i ny aa pau" ("i-ny") )
( oro_0311 "pau t aa t i ph aa pau" ("i-ph") )
( oro_0328 "pau t aa t o w aa pau" ("o-w") )
( oro_0329 "pau t aa t o x aa pau" ("o-x") )
( oro_0330 "pau t aa t o y aa pau" ("o-y") )
( oro_0331 "pau t aa t o z aa pau" ("o-z") )
( oro_0332 "pau t aa t o ch aa pau" ("o-ch") )
( oro_0333 "pau t aa t o dh aa pau" ("o-dh") )
( oro_0334 "pau t aa t o ny aa pau" ("o-ny") )
( oro_0335 "pau t aa t o ph aa pau" ("o-ph") )
( oro_0336 "pau t aa t o sh aa pau" ("o-sh") )
( oro_0345 "pau t aa t u l aa pau" ("u-l") )
( oro_0346 "pau t aa t u m aa pau" ("u-m") )
( oro_0347 "pau t aa t u n aa pau" ("u-n") )
( oro_0348 "pau t aa t u q aa pau" ("u-q") )
( oro_0349 "pau t aa t u r aa pau" ("u-r") )
( oro_0350 "pau t aa t u s aa pau" ("u-s") )
( oro_0351 "pau t aa t u t aa pau" ("u-t") )
( oro_0352 "pau t aa t u w aa pau" ("u-w") )
```

```
( oro_0353 "pau t aa t u x aa pau" ("u-x") )
( oro_0379 "pau t aa t aa z aa pau" ("aa-z") )
( oro_0380 "pau t aa t aa ch aa pau" ("aa-ch") )
( oro_0381 "pau t aa t aa dh aa pau" ("aa-dh") )
( oro_0382 "pau t aa t aa ny aa pau" ("aa-ny") )
( oro_0383 "pau t aa t aa ph aa pau" ("aa-ph") )
( oro_0384 "pau t aa t aa sh aa pau" ("aa-sh") )
( oro_0385 "pau t aa t ee b aa pau" ("ee-b") )
( oro_0386 "pau t aa t ee c aa pau" ("ee-c") )
( oro_0387 "pau t aa t ee d aa pau" ("ee-d") )
( oro_0388 "pau t aa t ee f aa pau" ("ee-f") )
( oro_0389 "pau t aa t ee g aa pau" ("ee-g") )
( oro_0390 "pau t aa t ee h aa pau" ("ee-h") )
( oro_0391 "pau t aa t ee j aa pau" ("ee-j") )
( oro_0392 "pau t aa t ee k aa pau" ("ee-k") )
( oro_0393 "pau t aa t ee l aa pau" ("ee-l") )
( oro_0429 "pau t aa t ii dh aa pau" ("ii-dh") )
( oro_0430 "pau t aa t ii ny aa pau" ("ii-ny") )
( oro_0431 "pau t aa t ii ph aa pau" ("ii-ph") )
( oro_0432 "pau t aa t ii sh aa pau" ("ii-sh") )
( oro_0433 "pau t aa t oo b aa pau" ("oo-b") )
( oro_0434 "pau t aa t oo c aa pau" ("oo-c") )
( oro_0435 "pau t aa t oo d aa pau" ("oo-d") )
( oro_0436 "pau t aa t oo f aa pau" ("oo-f") )
( oro_0437 "pau t aa t oo g aa pau" ("oo-g") )
( oro_0438 "pau t aa t oo h aa pau" ("oo-h") )
( oro_0439 "pau t aa t oo j aa pau" ("oo-j") )
( oro_0440 "pau t aa t oo k aa pau" ("oo-k") )
( oro_0512 "pau t aa f e t aa pau" ("f-e") )
( oro_0513 "pau t aa f i t aa pau" ("f-i") )
( oro_0514 "pau t aa f o t aa pau" ("f-o") )
( oro_0515 "pau t aa f u t aa pau" ("f-u") )
( oro_0516 "pau t aa f aa t aa pau" ("f-aa") )
( oro_0517 "pau t aa f ee t aa pau" ("f-ee") )
( oro_0518 "pau t aa f ii t aa pau" ("f-ii") )
( oro_0519 "pau t aa f oo t aa pau" ("f-oo") )
( oro_0520 "pau t aa f uu t aa pau" ("f-uu") )
( oro_0521 "pau t aa g a t aa pau" ("g-a") )
( oro_0522 "pau t aa g e t aa pau" ("g-e") )
( oro_0523 "pau t aa g i t aa pau" ("g-i") )
( oro_0524 "pau t aa g o t aa pau" ("g-o") )
( oro_0731 "pau t aa t e a t aa pau" ("e-a") )
( oro_0732 "pau t aa t e e t aa pau" ("e-e") )
( oro_0733 "pau t aa t e i t aa pau" ("e-i") )
( oro_0734 "pau t aa t e o t aa pau" ("e-o") )
( oro_0735 "pau t aa t e u t aa pau" ("e-u") )
( oro_0741 "pau t aa t i a t aa pau" ("i-a") )
( oro_0742 "pau t aa t i e t aa pau" ("i-e") )
( oro_0743 "pau t aa t i i t aa pau" ("i-i") )
( oro_0744 "pau t aa t i o t aa pau" ("i-o") )
( oro_0745 "pau t aa t i u t aa pau" ("i-u") )
```

F: Sample Afaan Oromo language Triphone List

```
( oro_0001 "t aa pau a b b aa pau" ("a-b" "b-b") )
( oro_0002 "t aa pau a c c aa pau" ("a-c" "c-c") )
( oro_0003 "t aa pau a d d aa pau" ("a-d" "d-d") )
( oro_0004 "t aa pau a f f aa pau" ("a-f" "f-f") )
( oro_0005 "t aa pau a g g aa pau" ("a-g" "g-g") )
( oro_0006 "t aa pau a h h aa pau" ("a-h" "h-h") )
( oro_0007 "t aa pau a j j aa pau" ("a-j" "j-j") )
( oro_0008 "t aa pau a k k aa pau" ("a-k" "k-k") )
( oro_0009 "t aa pau a l l aa pau" ("a-l" "l-l") )
( oro_0010 "t aa pau a m m aa pau" ("a-m" "m-m") )
( oro_0011 "t aa pau a n n aa pau" ("a-n" "n-n") )
( oro_0012 "t aa pau a q q aa pau" ("a-q" "q-q") )
( oro_0013 "t aa pau a r r aa pau" ("a-r" "r-r") )
( oro_0014 "t aa pau a s s aa pau" ("a-s" "s-s") )
( oro_0015 "t aa pau e b b aa pau" ("e-b" "b-b") )
( oro_0016 "t aa pau e c c aa pau" ("e-c" "c-c") )
( oro_0017 "t aa pau e d d aa pau" ("e-d" "d-d") )
( oro_0018 "t aa pau e f f aa pau" ("e-f" "f-f") )
( oro_0019 "t aa pau e g g aa pau" ("e-g" "g-g") )
( oro_0020 "t aa pau e h h aa pau" ("e-h" "h-h") )
( oro_0021 "t aa pau e j j aa pau" ("e-j" "j-j") )
( oro_0022 "t aa pau e k k aa pau" ("e-k" "k-k") )
( oro_0023 "t aa pau e l l aa pau" ("e-l" "l-l") )
( oro_0024 "t aa pau e m m aa pau" ("e-m" "m-m") )
( oro_0281 "pau t aa t a a b t aa pau" ("a-a" "a-b") )
( oro_0282 "pau t aa t a a c t aa pau" ("a-a" "a-c") )
( oro_0283 "pau t aa t a a d t aa pau" ("a-a" "a-d") )
( oro_0284 "pau t aa t a a f t aa pau" ("a-a" "a-f") )
( oro_0285 "pau t aa t a a g t aa pau" ("a-a" "a-g") )
( oro_0286 "pau t aa t a a h t aa pau" ("a-a" "a-h") )
( oro_0287 "pau t aa t a a j t aa pau" ("a-a" "a-j") )
( oro_0288 "pau t aa t a a k t aa pau" ("a-a" "a-k") )
( oro_0289 "pau t aa t a a l t aa pau" ("a-a" "a-l") )
( oro_0290 "pau t aa t a a m t aa pau" ("a-a" "a-m") )
( oro_0291 "pau t aa t a a n t aa pau" ("a-a" "a-n") )
( oro_0292 "pau t aa t a a q t aa pau" ("a-a" "a-q") )
( oro_0293 "pau t aa t a a r t aa pau" ("a-a" "a-r") )
( oro_0294 "pau t aa t a a s t aa pau" ("a-a" "a-s") )
( oro_0295 "pau t aa t e e b t aa pau" ("e-e" "e-b") )
( oro_0296 "pau t aa t e e c t aa pau" ("e-e" "e-c") )
( oro_0297 "pau t aa t e e d t aa pau" ("e-e" "e-d") )
( oro_0298 "pau t aa t e e f t aa pau" ("e-e" "e-f") )
( oro_0299 "pau t aa t e e g t aa pau" ("e-e" "e-g") )
( oro_0300 "pau t aa t e e h t aa pau" ("e-e" "e-h") )
( oro_0301 "pau t aa t e e j t aa pau" ("e-e" "e-j") )
( oro_0302 "pau t aa t e e k t aa pau" ("e-e" "e-k") )
( oro_0303 "pau t aa t e e l t aa pau" ("e-e" "e-l") )
```

```
( oro_0304 "pau t aa t e e m t aa pau" ("e-e" "e-m") )
( oro_0305 "pau t aa t e e n t aa pau" ("e-e" "e-n") )
( oro_0306 "pau t aa t e e q t aa pau" ("e-e" "e-q") )
( oro_0307 "pau t aa t e e r t aa pau" ("e-e" "e-r") )
( oro_0308 "pau t aa t e e s t aa pau" ("e-e" "e-s") )
( oro_0309 "pau t aa t i i b t aa pau" ("i-i" "i-b") )
( oro_0310 "pau t aa t i i c t aa pau" ("i-i" "i-c") )
( oro_0311 "pau t aa t i i d t aa pau" ("i-i" "i-d") )
( oro_0312 "pau t aa t i i f t aa pau" ("i-i" "i-f") )
( oro_0313 "pau t aa t i i g t aa pau" ("i-i" "i-g") )
( oro_0314 "pau t aa t i i h t aa pau" ("i-i" "i-h") )
( oro_0315 "pau t aa t i i j t aa pau" ("i-i" "i-j") )
( oro_0316 "pau t aa t i i k t aa pau" ("i-i" "i-k") )
( oro_0317 "pau t aa t i i l t aa pau" ("i-i" "i-l") )
( oro_0318 "pau t aa t i i m t aa pau" ("i-i" "i-m") )
( oro_0319 "pau t aa t i i n t aa pau" ("i-i" "i-n") )
( oro_0320 "pau t aa t i i q t aa pau" ("i-i" "i-q") )
( oro_0321 "pau t aa t i i r t aa pau" ("i-i" "i-r") )
( oro_0322 "pau t aa t i i s t aa pau" ("i-i" "i-s") )
( oro_0323 "pau t aa t o o b t aa pau" ("o-o" "o-b") )
( oro_0324 "pau t aa t o o c t aa pau" ("o-o" "o-c") )
( oro_0325 "pau t aa t o o d t aa pau" ("o-o" "o-d") )
( oro_0326 "pau t aa t o o f t aa pau" ("o-o" "o-f") )
( oro_0327 "pau t aa t o o g t aa pau" ("o-o" "o-g") )
( oro_0328 "pau t aa t o o h t aa pau" ("o-o" "o-h") )
( oro_0329 "pau t aa t o o j t aa pau" ("o-o" "o-j") )
( oro_0330 "pau t aa t o o k t aa pau" ("o-o" "o-k") )
( oro_0331 "pau t aa t o o l t aa pau" ("o-o" "o-l") )
( oro_0332 "pau t aa t o o m t aa pau" ("o-o" "o-m") )
( oro_0333 "pau t aa t o o n t aa pau" ("o-o" "o-n") )
( oro_0334 "pau t aa t o o q t aa pau" ("o-o" "o-q") )
( oro_0335 "pau t aa t o o r t aa pau" ("o-o" "o-r") )
( oro_0336 "pau t aa t o o s t aa pau" ("o-o" "o-s") )
( oro_0337 "pau t aa t u u b t aa pau" ("u-u" "u-b") )
( oro_0338 "pau t aa t u u c t aa pau" ("u-u" "u-c") )
( oro_0339 "pau t aa t u u d t aa pau" ("u-u" "u-d") )
( oro_0340 "pau t aa t u u f t aa pau" ("u-u" "u-f") )
( oro_0341 "pau t aa t u u g t aa pau" ("u-u" "u-g") )
( oro_0342 "pau t aa t u u h t aa pau" ("u-u" "u-h") )
( oro_0343 "pau t aa t u u j t aa pau" ("u-u" "u-j") )
( oro_0344 "pau t aa t u u k t aa pau" ("u-u" "u-k") )
( oro_0345 "pau t aa t u u l t aa pau" ("u-u" "u-l") )
( oro_0346 "pau t aa t u u m t aa pau" ("u-u" "u-m") )
( oro_0347 "pau t aa t u u n t aa pau" ("u-u" "u-n") )
( oro_0348 "pau t aa t u u q t aa pau" ("u-u" "u-q") )
( oro_0349 "pau t aa t u u r t aa pau" ("u-u" "u-r") )
( oro_0350 "pau t aa t u u s t aa pau" ("u-u" "u-s") )
```

G: Sentences Used to Test the Afaan Oromo language TTS system

1. NAGAA, GALATA WAAQAYYO

   →I am fine, Praise the Lord

2. HOJII ISAA NATTI HIMTAA?

   →Can you tell me what he is doing?

3. NYAATA BIRAA HIN BARBAADU

   →I don't need more food

4. ODUU GAARII

   →Good news

5. BAGA NAGAAN TURTAN

   →Well stay

6. ATI BANTUU KEE NI GATTE

   →You lost your key

H: Words Used to Test the Afaan Oromo language TTS

No.	Words	The English Equivalent
1	Abjuu	Dream
2	Afaan	Language
3	Afeeruu	To invite
4	Affeeluu	To cook
5	Afuura	Breath
6	Agarsiisuu	To show
7	Ajjeesuu	To kill
8	Baadiyaa	Rural area
9	Baala	Leaf
10	Baayyee	Many
11	Bantuu	Key
12	Bareedduu	Beautiful
12	Bilbila	Telephone
13	Bishaan	Water
14	Biyyaa	Country
15	Bu'aa	Profit
16	Cunqursaa	Oppression
17	Dabarsuu	Transfer
18	Dafee	Early
19	Dandeettii	Honey
20	Dhaba	Poverty
21	Dhadhaa	Butter
22	Dhalaa	Female
23	Doofaa	Illitrate
24	Durdur	Long ago
25	Duwwaa	Empty
26	Ergaa	Message
27	Fakkeenya	Example
28	Farda	Horse
29	Fira	Relative
30	Fuudha	Marriage
31	Gaarii	Good

32	Gabaa	Market
33	Gadhee	Bad
34	Galgalaa	Evening
35	Gammachuu	Pleasure
36	Gammachuu	Joy
37	Ganda	Village
38	Garaagar	Different
39	Garba	Slaves
40	Gatte	Lost
41	Guddaa	Big
42	Gumaata	Gift
43	Haaraa	New
44	Harka	Hand
45	Hattuu	Thieves
46	Ibidda	Fire
47	Iccitii	Secret
48	Irbuu	Promise
49	Kaleessa	Yesterday
50	Laga	River
51	Mallattoo	Signature
52	Mana	House
53	Meeqa	How much
54	Michuu	Friend
55	Nagaa	Peace
56	Nyaata	Food
57	Oduu	News
58	Qabeenya	Wealth
59	Qarshii	Money
60	Qawwee	Gun
61	Qaxuuxee	Thin
62	Qirixuu	To cut
63	Qullubbii	Onion
64	Qulqullu	Pure
65	Rakkoo	Difficult
66	Rifeensa	Hair
67	Rifeensa	Hair
68	Rooba	Rain

69	Sangaa	Ox
70	Shaakala	Practise
71	Sirrii	Right
72	Sooressa	Rich
73	Suuta	Slowly
74	Teessoo	Address
75	Tumaa	Regulation
76	Umurii	Age
77	Urjii	Stars
78	Uummata	People
79	Waggaa	Year
80	Wangeela	Gospel
81	Warshallee	Factory
82	Xinnoo	Little
83	Xiyyaara	Air Port
84	Yaallii	Try